US009333645B1

(12) **United States Patent**
Wu et al.

(10) **Patent No.:** **US 9,333,645 B1**
(45) **Date of Patent:** **May 10, 2016**

(54) **OPTIMIZATION OF START-UP TRANSIENT PROCESSES FOR DUAL-ARMED CLUSTER TOOLS WITH WAFER REVISITING**

(71) Applicant: **Macau University of Science and Technology**, Macau (MO)

(72) Inventors: **Naiqi Wu**, Mo (MO); **Mengchu Zhou**, Newark, NJ (US); **Chunrong Pan**, Mo (MO); **Yan Qiao**, Mo (MO)

(73) Assignee: **Macau University Science and Technology**, Macau (MO)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **14/639,960**

(22) Filed: **Mar. 5, 2015**

Related U.S. Application Data

(60) Provisional application No. 62/102,111, filed on Jan. 12, 2015.

(51) **Int. Cl.**

G05D 3/12	(2006.01)
B25J 9/00	(2006.01)
G05B 19/418	(2006.01)
G06F 17/50	(2006.01)

(52) **U.S. Cl.**
CPC ***B25J 9/0087*** (2013.01); ***G05B 19/41865*** (2013.01); ***G06F 17/5013*** (2013.01); *G05B 2219/45031* (2013.01); *Y10S 901/02* (2013.01); *Y10S 901/41* (2013.01)

(58) **Field of Classification Search**
CPC B25J 9/0087; G06F 17/5013; G05B 19/41865; G05B 2219/45031; Y10S 901/41; Y10S 901/02
USPC 700/298
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,418,350	B1 *	7/2002	Hamidzadeh	G05B 19/41865 700/100
7,451,011	B2 *	11/2008	Funk	H01L 22/20 257/E21.525
2004/0078109	A1 *	4/2004	Babikian	G05B 19/41865 700/121
2011/0180148	A1 *	7/2011	Xia	B01F 3/04106 137/1

OTHER PUBLICATIONS

Petri Net-based scheduling analysis of dual-arm cluseter tools with wafer revisiting by Qiao et al. dated Aug. 20-24, 2012.*

Petri Net-based cycle time analyis of dual-arm cluster tools with wafer revisiting and swapping strategy, by Wu et al. dated May 9-13, 2011.*

Petri Net-based scheduling analysis of dual-arm cluseter tools subject to wafer revisiting and residency time constraints by Qiao et al. dated Apr. 10-12, 2013.*

Scheduling and Analysis of Start-Up Transient Processes for Dual-Arm Cluster Tools With Wafer Revisiting by Pan et al. dated Jan. 12, 2015.*

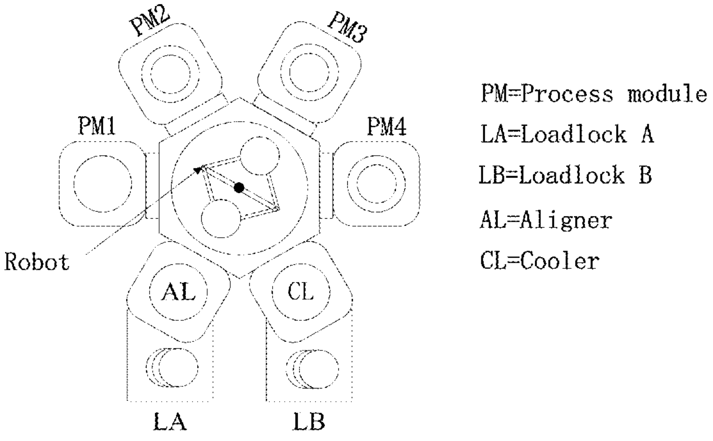
* cited by examiner

Primary Examiner — Mohammad Ali
Assistant Examiner — Ziaul Karim
(74) *Attorney, Agent, or Firm* — Ella Cheong Hong Kong; Sam T. Yip

(57) **ABSTRACT**

A method for scheduling dual-armed cluster tools with wafer revisiting is provided. In order to speed up start-up transient processes, the present invention adopts a program evaluation and review technique for the analysis of start-up transient processes and develops optimization algorithms for their scheduling for dual-arm cluster tools. Then, their complexity is analyzed.

8 Claims, 6 Drawing Sheets



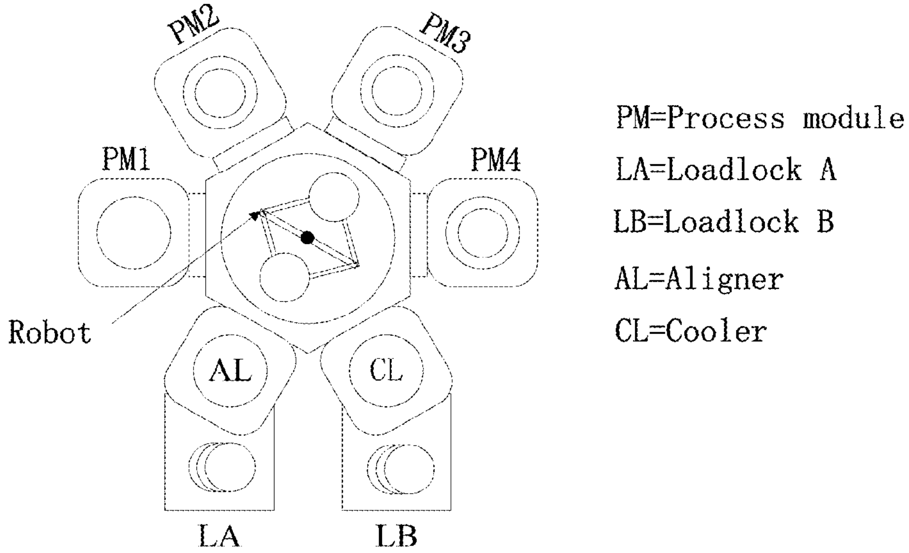


FIG. 1

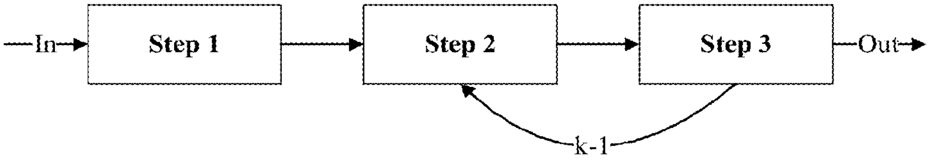


FIG. 2A

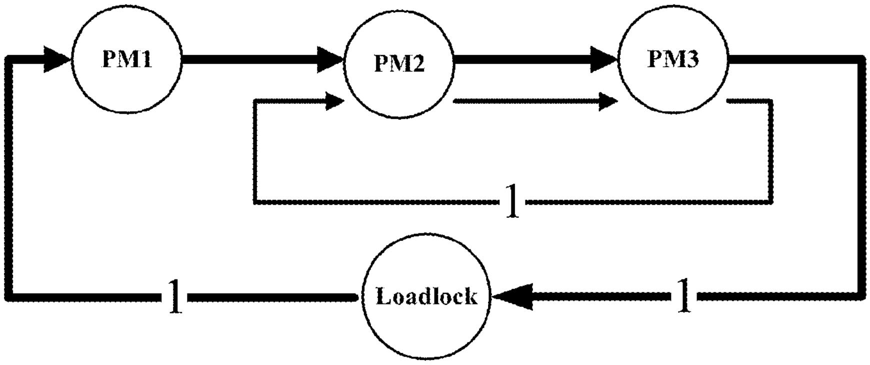


FIG. 2B

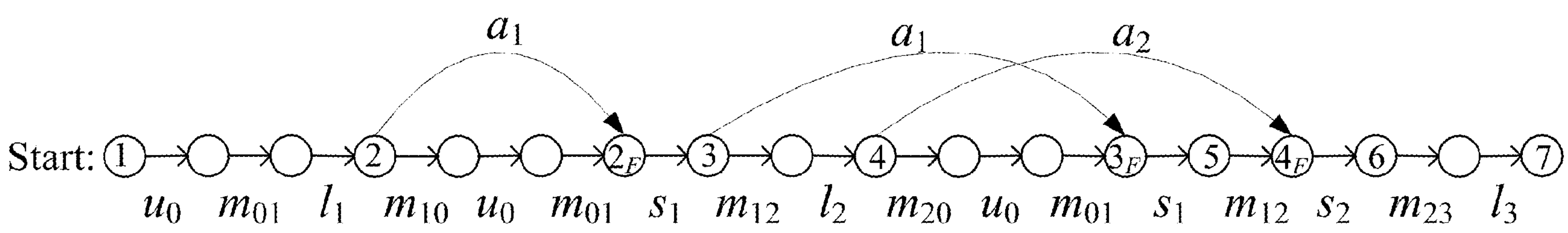

a_1
a_1
a_2
Start:
1
2
2_F
3
4
3_F
5
4_F
6
7
u_0
m_{01}
l_1
m_{10}
u_0
m_{01}
s_1
m_{12}
l_2
m_{20}
u_0
m_{01}
s_1
m_{12}
s_2
m_{23}
l_3

FIG. 3A

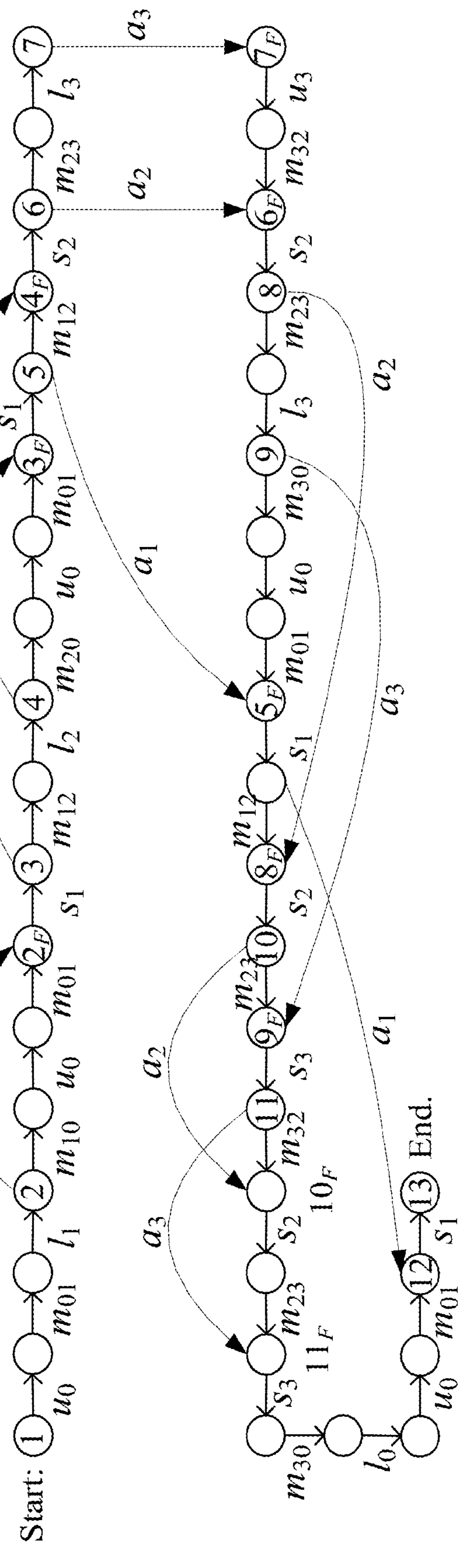

Start:
1
u_0
m_{01}
l_1
2
m_{10}
u_0
m_{01}
2_F
s_1
3
m_{12}
l_2
4
m_{20}
u_0
m_{01}
3_F
s_1
5
m_{12}
4_F
s_2
6
m_{23}
l_3
7
a_1
a_1
a_2
a_1
a_2
a_3
7_F
u_3
m_{32}
6_F
s_2
8
m_{23}
l_3
9
m_{30}
u_0
m_{01}
5_F
s_1
m_{12}
8_F
s_2
10
m_{23}
9_F
s_3
11
m_{32}
10_F
s_2
m_{23}
11_F
s_3
m_{30}
l_0
u_0
m_{01}
12
s_1
13
End.
a_2
a_3
a_1
a_2
a_3

FIG. 3B

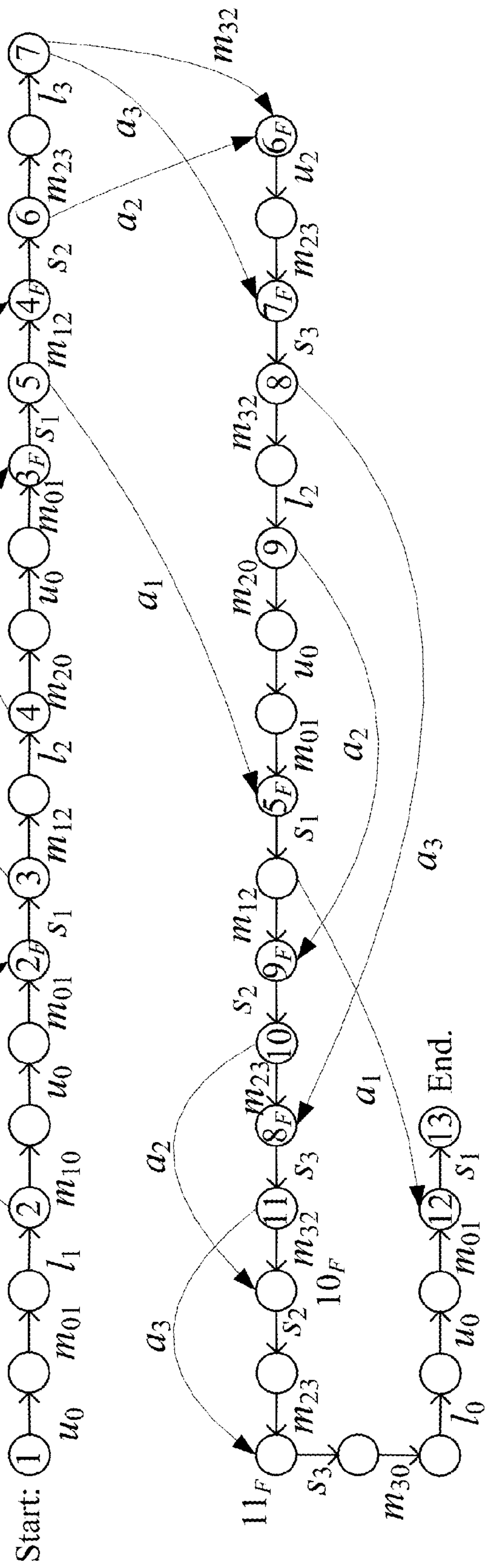

Start:
1
u_0
m_{01}
l_1
2
m_{10}
u_0
m_{01}
2_F
s_1
3
m_{12}
l_2
4
m_{20}
u_0
m_{01}
3_F
s_1
5
m_{12}
4_F
s_2
6
m_{23}
l_3
7
a_1
a_1
a_2
a_1
a_2
a_3
m_{32}
6_F
u_2
m_{23}
7_F
s_3
8
m_{32}
l_2
9
m_{20}
u_0
m_{01}
5_F
s_1
m_{12}
9_F
s_2
10
m_{23}
8_F
s_3
11
m_{32}
10_F
s_2
m_{23}
11_F
s_3
m_{30}
l_0
u_0
m_{01}
12
s_1
13
End.
a_2
a_3
a_2
a_3
a_1

FIG. 3C

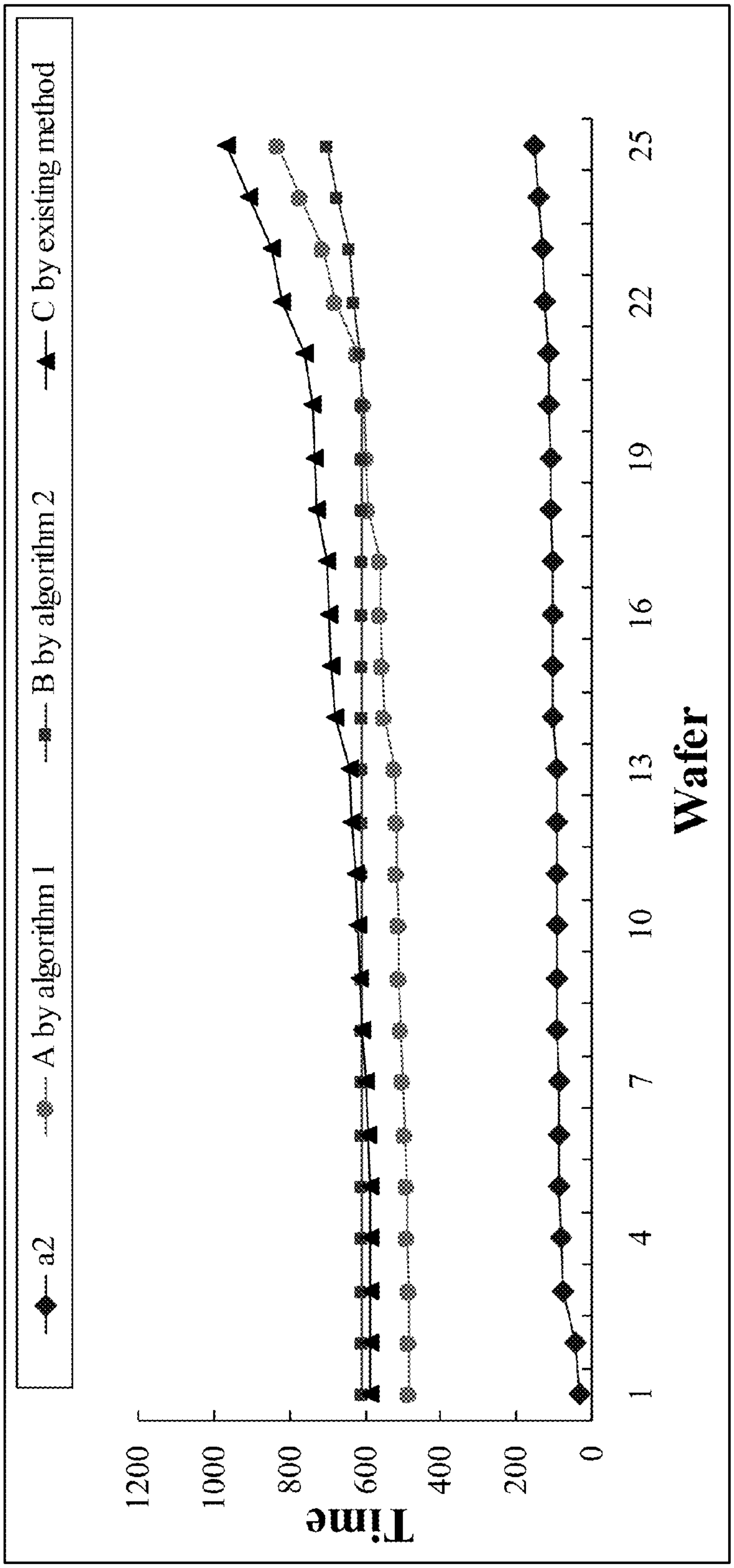

a2
A by algorithm 1
B by algorithm 2
C by existing method
Time
1200
1000
800
600
400
200
0
Wafer
1
4
7
10
13
16
19
22
25

FIG. 4A

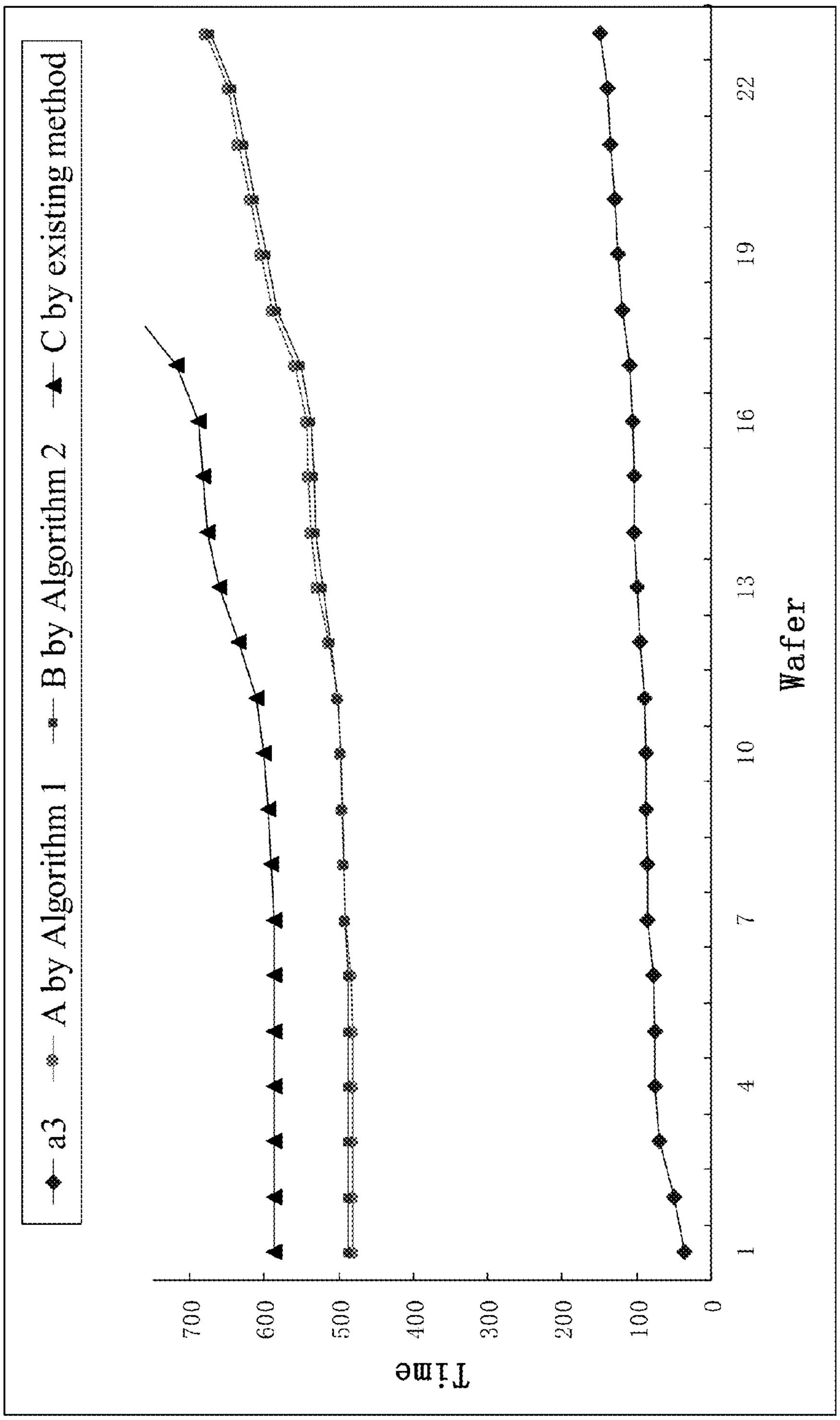

a3
A by Algorithm 1
B by Algorithm 2
C by existing method
Time
0
100
200
300
400
500
600
700
Wafer
1
4
7
10
13
16
19
22

FIG. 4B

OPTIMIZATION OF START-UP TRANSIENT PROCESSES FOR DUAL-ARMED CLUSTER TOOLS WITH WAFER REVISITING

CLAIM FOR DOMESTIC PRIORITY

This application claims priority under 35 U.S.C. §119 to the U.S. Provisional Patent Application No. 62/102,111 filed Jan. 12, 2015, the disclosure of which is incorporated herein by reference in its entirety.

COPYRIGHT NOTICE

A portion of the disclosure of this patent document contains material, which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

FIELD OF THE INVENTION

The present invention relates to a method for scheduling dual-armed cluster tools with wafer revisiting.

BACKGROUND

The following references are cited in the specification. Disclosures of these references are incorporated herein by reference in their entirety.

List of References:

Y. Ahn and J. R. Morrison, "Analysis of circular cluster tools: Transient behavior and semiconductor equipment models," in *Proceedings of the* 2010 *IEEE Conference on Automation Science and Engineering*, Toronto, Canada, pp. 39-44, August 2010.

M. Bader, R. Hall and G. Strasser, "Integrated processing equipment," *Solid State Technog.*, vol.33, no.5, pp.149-154, 1990.

P. Burggraaf, "Coping with the high cost of wafer fabs." *Semiconductor International*, vol.38, pp. 45-50, 1995.

W. K. V. Chan, J. Yi, and S. Ding, "Optimal scheduling of multicluster tools with constant robot moving times: two-cluster analysis," *IEEE Transactions on Automation Science and Engineering*, vol.8 no. 1, pp. 5-16, January 2011.

S. Ding, J. Yi, and M. Zhang, "Scheduling multi-cluster tools: An integrated event graph and network model approach," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 3, pp. 339-351, August 2006.

J. H. Lee, H. J. Kim, and T. E. Lee, "Scheduling transient periods of dual-armed cluster tools," in *Proceedings of the* 2012 *IEEE International Conference on Mechatronics and Automation*, Chengdu, China, pp. 1569-1574, August 2012.

D. J. Kim, E. Cimren, R. Havery, and A. K. Zaidi, "Improving cluster tools performance using colored Petri nets", Semiconductor Manufacturing," *Winter simulation conference*: 1-205. 2012.

T. K. Kim, C. Jung and T. E. Lee, "Scheduling start-up and close-down periods of dual-armed cluster tools with wafer delay regulation" *International Journal of Production Research*, vol. 50, no. 10, pp. 2785-2795, May 2012.

H. J. Kim, J. H. Lee, C. Jung, and T. E. Lee, "Scheduling cluster tools with ready time constraints for consecutive small lots," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 1, pp. 145-159, January 2013a.

T. K. Kim, C. Jung, and T. E. Lee, "Scheduling start-up and closedown periods of dual-armed cluster tools with wafer delay regulation," *International Journal of Production Research*, vol. 50, no. 10, pp. 2785-2795, May 2013b.

H. Y. Lee and T. E. Lee, "Scheduling single-arm cluster tools with reentrant wafer flows," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 2, pp. 226-240, May 2006.

T. L. Perkinson, P. K. Maclarty, R. S. Gyurcsik, and R. K. Cavin, III, "Single-wafer cluster tools performance: An analysis of throughput," *IEEE Transactions on Semiconductor Manufacturing*, vol. 7, no. 2, pp. 369-373, May 1994.

T. L. Perkinston, R. S. Gyurcsik, and P. K. Maclarty, "Single-wafer cluster tool performance: An analysis of effects of redundant chambers and revisitation sequences on throughput," *IEEE Transactions on Semiconductor Manufacturing*, vol. 9, no. 2, pp. 384-400. May 1996.

Yan, Qiao, N. Q. Wu, and M. C. Zhou, "A Petri Net-Based Novel Scheduling Approach and its Cycle Time Analysis for Dual-Arm cluster Tools with Wafer Revisiting" *IEEE Transactions on Semiconductor Manufacturing*, vol.26, no.1, pp. 100-110, February 2013.

S. Venkatesh, R. Davenport, P. Foxhoven, and J. Nulman, "A steady state throughput analysis of cluster tools: Dual-blade versus single-blade robots," *IEEE Transactions on Semiconductor Manufacturing*, vol. 10, no. 4, pp. 418-424, November 1997.

N. Q. Wu and M. C. Zhou, "Colored time Petri nets for modeling and analysis of cluster tools," *Asian Journal of Control*, vol. 12, no. 3, pp. 253-266, 2010.

N. Q. Wu, C. B. Chu, F. Chu, and M. C. Zhou, "Petri net modeling and cycle time analysis of dual-arm cluster tools with wafer revisiting," *IEEE Transaction on Systems Man Cybernetics, Part A*, vol. 43, no. 1, 196-207, January 2013a.

N. Q. Wu and M. C. Zhou, F. C. Chu and C. B. Chu, "A Petri-net-based scheduling strategy for dual-arm cluster tools with wafer revisiting," *IEEE Transaction on Systems Man Cybernetics, Part A*, vol. 43, no. 5, 1182-1194, September 2013b

U. Wikborg and T. E. Lee, "Noncyclic scheduling for timed discrete event systems with application to single-armed cluster tools using Pareto-optimal optimization," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 689-710, July 2013.

J. Yi, S. Ding, and M. Zhang, "Steady-state throughput and scheduling analysis of multi-cluster tools: A decomposition approach," *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 2, pp. 321-336, April 2008.

W. M. Zuberek, "Timed Petri nets in modeling and analysis of cluster tools," *IEEE Transaction on Robotics & Automation Magazine*, vol. 17, no. 5, pp.562-575, October 2001.

W. M. Zuberek, "Cluster tools with chamber revisiting: modeling and analysis using time Petri nets," *IEEE Transactions on Semiconductor Manufacturing*, vol. 17, no. 3, pp. 333-344, August 2004.

To achieve better quality control and reduce lead time, cluster tools are widely used for wafer processing in semiconductor manufacturing systems with a single-wafer processing technology that processes wafer one by one at a process module (PM). They provide a reconfigurable, flexible and efficient environment [Bader et al., 1990; and Burggraaf, 1995]. A cluster tool consists of several PMs, an aligner module, a wafer handling robot (transfer module), and loadlocks (LL) for wafer cassette loading/unloading, which are mechanically linked together in a radial way and computercontrolled. With a single-arm or dual-arm robot, such tool is correspondingly called a single-arm or dual-arm cluster tool. The latter is shown in FIG. **1**.

A cluster tool with two LLs can be operated consecutively without being interrupted. Thus, it can operate in a steady state for most of time. Most prior studies [Chan et al., 2011; Ding et al., 2006; Perkinston et al., 1994; Perkinston et al., 1996; Venkatesh et al., 1997; Wu and Zhou, 2010; Yi et al., 2008; and Zuberek, 2001] were conducted to find an optimal periodical schedule. Only limited researches focused on transient states [Kim et al., 2013a; Kim et al., 2013b; Lee et al., 2012; Ahn and Morrison, 2010; and Wikborg and Lee, 2013] despite their increasing importance. With a given robot task sequence, the transient cycle time of a dual-arm cluster tool is minimized [Kim et al., 2013b]. Deadlock-free conditions are presented for lot switching periods of dual-arm cluster tools [Lee et al., 2012]. A model for transient and steady states is built [Ahn and Morrison, 2010]. When a robot task sequence is determined, an algorithm for an optimal transient cycle is proposed [Kim et al., 2013a, and Wikborg and Lee, 2013]. Since frequent switches between transient and steady states may lead to a deadlock problem, optimizing transient processes for cluster tools is recognized to be a big research challenge.

Some wafer fabrication processes, for instance, atomic layer deposition (ALD), require a wafer to visit more than once a PM under identical processing conditions. Due to such revisits, wafer processes are no longer a flow-shop, and the prior results without considering revisits, e.g. [Kim et al., 2013(a); and Wikborg and Lee, 2013], are no longer applicable. Such revisits make the optimal scheduling of transient processes much more challenging. Note that some well-developed theories or rules for a steady process [Lee and Lee, 2006] are not applicable for scheduling transient processes. Furthermore, deadlock often becomes an issue in a wafer fabrication process with revisit in a cluster tool [Lee and Lee, 2006]. It is certainly much more challenging to schedule a deadlock-prone system to obtain an optimal and deadlock-free schedule. In order to avoid deadlock and minimize makespan for a dual-arm cluster tool with wafer revisit, the studies [Wu and Zhou, 2010; and Zuberek, 2004] have developed Petri net models for performance evaluation and analyzed its cycle time under a swap strategy. They show that the steady state cycle time is determined by the time taken for completing one wafer at the step that has the heaviest workload. In fact, it is the lower bound of the cycle time for the system. However, this result is shown to be not correct in [Wu et al., 2013b], where a swap-based strategy is proposed and leads to a three-wafer cyclic schedule, which includes three local cycles for a revisiting process and three global cycles. It is also shown that the system can never enter its steady state under some conditions. In other words, the steady state-based analysis methods given in [Wu and Zhou, 2010; and Zuberek, 2004] are not applicable and the tool cannot reach the lower bound of cycle time. The multiple local and global cycles reduce the productivity of such cluster tools [Wu et al., 2013a]. Can a schedule with fewer local and global cycles improve the performance? The work [Wu et al., 2013b; and Qiao et al., 2013] answers this question by providing a schedule with two-wafer cycle, and also the one with one cycle only. It finds that the less the number of global cycles is, the shorter the robot waiting time is. Qiao et al. [2013] prove that one-wafer periodic schedule is optimal. They also provide a novel Petri net-based method to evolve a system from a transient state to a steady one. However, the transient process is not optimal despite its easy implementation according to the results to be presented in this work.

Although the minimum transient period is proposed for transient process scheduling [Kim et al., 2012], such conclusion is applicable to a non-revisiting case only. Their method cannot be extended a process with revisits without substantial research. The transient scheduling of dual-arm cluster tools with wafer revisiting to reach a desired steady state optimally is widely open as one has not found any study addressing it.

There is a need in the art for a method to obtain a schedule for dual-armed cluster tools with wafer revisiting process.

SUMMARY OF THE INVENTION

An aspect of the present invention is to provide a method for scheduling dual-armed cluster tools with wafer revisiting process under optimization of start-up transient processes.

According to an embodiment of the present claimed invention, a method for scheduling a dual-armed cluster tools with wafer revisiting, comprises: obtaining, by a processor, a wafer processing time, a wafer loading time, a wafer unloading time, a moving time from one process module to another, and a swapping time; determining, by a processor, one or more algorithms of transient process based on a program evaluation and review technique; computing, by a processor, a time to reach steady state via the algorithms based on the wafer processing time, the wafer loading time, the wafer unloading time, the moving time, and the swapping time; and determining, by a processor, a scheduling strategy given by the algorithm of transient process having a minimal time to reach steady state.

In order to speed up start-up transient processes, the present invention adopts a program evaluation and review technique for the analysis of start-up transient processes and develops optimization algorithms for their scheduling for dual-arm cluster tools.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the present invention are described in more detail hereinafter with reference to the drawings, in which:

FIG. **1** shows a dual-arm cluster tool with four PMs;

FIG. **2**A depicts a wafer flow for ALD process;

FIG. **2**B depicts a wafer flow process by using one-wafer schedule;

FIG. **3**A depicts the PERT from M_0 to M_3;

FIG. **3**B depicts the PERT for Case 1;

FIG. **3**C depicts the PERT for Case 2;

FIG. **4**A depicts variation trend of A, B and C as a_2 increases; and

FIG. **4**B depicts variation trend of A, B and C as a_3 increases.

DETAILED DESCRIPTION

In the following description, a method for scheduling dual-armed cluster tools with wafer revisiting process under optimization of start-up transient processes is set forth as preferred examples. It will be apparent to those skilled in the art that modifications, including additions and/or substitutions maybe made without departing from the scope and spirit of the invention. Specific details may be omitted so as not to obscure the invention; however, the disclosure is written to enable one skilled in the art to practice the teachings herein without undue experimentation.

The trends of increasing wafer diameter and smaller lot sizes have led to more transient periods in wafer fabrication. For some wafer fabrication processes, such as atomic layer deposition (ALD), wafers need to visit some process modules for a number of times, instead of once, thus leading to a so-called revisiting process. Most previous studies on cluster tool scheduling focus on steady periodic cycle, in which cluster tools repeat identical cycles. Research on transient processes of dual-arm cluster tools with wafer revisiting processes becomes urgently needed for high-performance wafer fabrication. In order to speed up start-up transient processes, the present invention adopts a program evaluation and review technique for the analysis of start-up transient processes and develops optimization algorithms for their scheduling for dual-arm cluster tools. Then, their complexity is analyzed. Finally, illustrative examples are given to show the applications of the proposed method.

Section A briefly introduces the processes and one-wafer scheduling method. Section B develops start-up transient process scheduling and optimization algorithms. Section C presents illustrative examples.

A. REVISING PROCESS AND ONE-WAFER SCHEDULING METHOD

A.1 Atomic Layer Deposition Process

As for wafer fabrication, there are many operations for processing a wafer. If some operations require the same fabrication process as it visits its prior steps [Chan et al., 2011], it is called revisiting. If every operation of a wafer needs different fabrication processes, it is called non-revisiting. A revisiting process may contain only one step or more. A two-step revisiting process is typical and thus considered as an example in the present invention. For an ALD process, the thickness of the deposition layer is determined by the number of revisiting times. In an ALD process, there are three steps, and a wafer visits Step 1 once, and then Steps 2 and 3 for k≥2 times as shown in FIG. 2A, where k is determined by a process plan. One assumes that PM_1-PM_3 are configured for Steps 1-3, respectively. Then, one describes a wafer flow pattern of an ALD process as $(PM_1, (PM_2, PM_3)^k)$ with $(PM_2, PM_3)^k$ being a k-revisiting process. Without loss of generality and to make the present invention easy to follow, one assumes that k=2, i.e., $(PM_1, (PM_2, PM_3)^2)$.

A.2 Activity Description

There are several processing steps in a cluster tool for wafer fabrication. LLs can also be seen as a wafer processing step. Hence, one treats LLs as Step 0. According to [Shin et al., 2001], in operation of a cluster tool, PM activities follow the robot tasks. Thus, it is critically important to schedule robot activities. They include unloading a wafer from a PM, moving from a PM to another with a wafer carried, loading a wafer into a PM, moving from a PM to another without carrying a wafer, and waiting. The key in operating a cluster tool is to schedule the robot activities given the tool's status and process requirements. One uses u_i and l_i to denote the robot unloading and loading a wafer from P_i, $i \in N_3=\{1, 2, 3\}$, respectively. As mentioned in Background, a swap strategy is efficient for scheduling dual-arm cluster tools. A swap operation at PM_i is executed as follows: the robot holds a wafer in one arm→unloads a processed wafer from a PM_i by the other arm→the robot rotates→loads a raw wafer into PM_i. In this way, a swap operation at PM_i is completed. It follows from this process that l_i and u_i together with a rotation form a swap operation at Step i, $i \in N_3$, and one uses s_i to denote it. One uses m_{ij} to denote the robot moving from Steps i to j. In the present invention, Steps 2 and 3 together form a revisiting process step. Therefore, m_{32} represents the robot moving from Steps 3 to 2.

For the purpose of scheduling, the temporal characterization for each activity is necessary. The time for the robot to unload a wafer is denoted as α. Similarly, the time for the robot to load a wafer to a PM and move from a PM to another are denoted as β and μ, respectively. They are listed in Table 2.1. Although a swap operation includes unloading and loading a wafer, the time of a swap operation is not simply their sum. One uses λ to denote the time. Besides the robot activities, one uses a_i to denote the wafer processing time at Step i, $i \in N_3$.

TABLE 2.1

The meanings of the denotations

Notations	Robot Tasks	Time
u_i	Unloading a wafer from Step i	α
l_i	Loading a wafer into Step i	β
m_{ij}	Moving from Steps i to j	μ
s_i	Swapping at Step i	λ

A.3. One Wafer Scheduling Strategy

One first presents how to obtain an optimal periodic schedule for the k=2 case. One defines m_{ij} for the robot moving from PM_i to PM_j. l_0 and u_0 are defined to load a wafer to LL and unload a wafer from LL for the robot, respectively. If the robot executes tasks in the revisiting process $(PM_2, PM_3)^2$ with a sequence $\sigma_1=\langle$ swapping at $PM_3 \rightarrow m_{32} \rightarrow$swapping at $PM_2 \rightarrow m_{23} \rangle$ that forms a cycle, one calls it a local cycle. If a robot executes tasks with a sequence $\sigma_2=\langle$ swapping at $PM_3 \rightarrow m_{30} \rightarrow l_0 \rightarrow u_0 \rightarrow m_{01} \rightarrow$swapping at $PM_1 \rightarrow m_{12} \rightarrow$ swapping at $PM_2 \rightarrow m_{23} \rangle$ that forms a cycle involving all PMs once, one calls it a global cycle. One-wafer schedule should contain one local and one global cycle as shown in FIG. 2B. It must be optimal in the sense of cycle time according to [Qiao et al., 2013]. Let $M=\{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4\}$ denote the state of the system, where $\Gamma_i=\{W_d(q)\}$, $i \in N_3=\{1, 2, 3\}$, with $W_d(q)$ being the d-th wafer released to the system with its q-th operation being processed in PM_i (Step i). $\Gamma_4=\{R_j(W_d(q)\}$ represents the d-th wafer that is held by the robot with its q-th operation to be processed at Step j, $j \in N_3$. What is the first desired state to guarantee such an optimal one-wafer steady-state schedule? The answer is that the tool starts from the state $M_i=\{W_3(1), W_1(4), W_2(3), R_1(W_4(1))\}$ according to [Qiao et al., 2013], which is called the first desired steady state. It represents that the 3rd, 1st and 2nd wafers are being processed in PM_1, PM_2, and PM_3 for their 1st, 4th, and 3rd operations, respectively. At the same time, the robot holds the 4-th wafer with the 1st operation to be processed at Step 1. At this state, by executing $\sigma_3=\langle$ swapping at $PM_1 \rightarrow m_{12} \rightarrow$swapping at $PM_2 \rightarrow m_{23} \rangle$, the system state becomes $M_{i+1}=\{W_4(1), W_3(2), W_2(3), R_3(W_1(5))\}$. By executing σ_1, it becomes $M_{i+2}=\{W_4(1), W_2(4), W_1(5), R_3(W_3(3))\}$. Finally, by executing $\sigma_4=\langle$ swapping at $PM_3 \rightarrow m_{30} \rightarrow l_0 \rightarrow u_0 \rightarrow m_{01} \rangle$, $M_{i+3}=\{W_4(1), W_2(4), W_3(3), R_1(W_5(1))\}$ is obtained. One can find that σ_3 and σ_4 together form a global cycle. In the meantime, M_i and M_{i+3} are equivalent. Therefore, a period including a local and a global cycle is formed. During this period, one wafer is unloaded from LL, while another is completed and returns to LL.

Next, one analyzes how the system can optimally enter the first desired steady state from the initial state.

B. START-UP TRANSIENT PROCESS SCHEDULING

Qiao et al. [2013] show that a one-wafer cyclic schedule is optimal for a dual-arm cluster tool with wafer revisiting in terms of cycle time. The first desired state $M_i=\{W_3(1), W_1(4), W_2(3), R_1(W_4(1))\}$ in such a cyclic schedule. Thus, the question to be answered is how to reach it from initial state $M_0=\{\Theta, \Theta, \Theta, R_0(\Theta)\}$ in the fastest way, where Θ means the robot arm is empty and ready for unloading a wafer from LL. To do so, one develops the following method.

Let ω_i^f denote waiting time for the robot to unload a wafer from a PM_i during the period from states M_{j-1} to M_j, j>0. At the initial state, there are no wafers being processed in PMs. Then, to reach $M_1=\{W_1(1), \Theta, \Theta, R_1(\Theta)\}$, the robot performs a sequence ⟨unloading a wafer W_1 from LL (u_0), moving to PM_1 (m_{01}), and loading the wafer to PM_1 (l_1)⟩. To reach $M_2=\{W_2(1), W_1(2), \Theta, R_2(\Theta)\}$, it performs ⟨moving to LL (m_{10}), unloading a wafer W_2 from LL (u_0), moving to PM_1 (m_{01}), waiting at PM_1 (ω_1^2), swapping at PM_1, moving to PM_2 (m_{12}), and loading wafer W_1 to PM_2 (l_2)⟩. To reach $M_3=\{W_3(1), W_2(2), W_1(3), R_3(\Theta)\}$, the robot performs ⟨moving to LL (m_{20}), unloading the wafer W_3 from LL (u_0), moving to PM_1 (m_{01}), waiting at PM_1 (ω_1^3), swapping at PM_1, moving to PM_2 (m_{12}), waiting at PM_2 (ω_2^3), swapping at PM_2, moving to PM_3 (m_{23}), and loading the wafer W_1 into PM_3 (l_3)⟩. Based on the above robot task sequence, one can obtain a program evaluation and review technique (PERT) chart shown in FIG. **3**A. As for Nodes 1, 2, 4 and 7, the system states are M_0, M_1, M_2 and M_3, respectively. Note that, from Node 2 to Node 2F, two concurrent sequences of operations must be finished, they are 1) robot tasks from PM_1 to LL and LL to PM_1 and 2) processing a wafer $W_1(1)$. From Node 3 to Node 3F, two concurrent sequences of operations must be finished, 1) robot tasks from PM_1 to PM_2, PM_2 to LL, and LL to PM_1 and 2) processing a wafer $W_2(1)$. From Node 4 to Node 4F, two concurrent sequences of operations must be finished, 1) robot tasks from PM_2 to LL, LL to PM_1, and PM_1 to PM_2 and 2) processing a wafer $W_1(2)$.

To promptly reach the first desired steady state, according to the process time difference between PM_2 and PM_3, there are two cases. For Case 1, the system evolves as shown in FIG. **3**B and states are as follows:

To reach $M_{41}=\{W_3(1), W_1(4), W_2(3), R_3(\Theta)\}$ at Node 9, the robot performs ⟨staying at PM_3 (ω_3^4), unloading wafer $W_1(3)$ from PM_3 (u_3), moving to PM_2 (m_{32}), waiting at PM_2 (ω_2^4), swapping at PM_2, moving to PM_3 (m_{23}), and loading wafer $W_2(3)$ into PM_3 (l_3)⟩;

To reach $M_{51}=\{W_3(1), W_1(4), W_2(3), R_1(W_4(1))\}$ at Node 5F, the robot performs ⟨moving to LL (m_{30}), unloading wafer $W_4(1)$ from LL (u_0), moving to PM_1 (m_{01}), and waiting at PM_1 (ω_1^5)⟩;

To reach $M_{61}=\{W_4(1), W_3(2), W_2(3), R_3(W_1(5))\}$ at Node 9F, the robot performs ⟨swapping at PM_1, moving to PM_2 (m_{12}), waiting at PM_2 (ω_2^6), swapping at PM_2, moves to PM_3 (m_{23}), and waiting at PM_3 (ω_3^6)⟩;

To reach $M_{71}=\{W_4(1), W_2(4), W_1(5), R_3(W_3(3))\}$ at Node 11F, the robot performs ⟨swapping at PM_3, moving to PM_2 (m_{32}), waiting at PM_2 (ω_2^7), swapping at PM_2, moving to PM_3 (m_{23}), and waiting at PM_3 (ω_3^7)⟩;

To reach $M_{81}=\{W_4(1), W_2(4), W_3(3), R_1(W_5(1))\}$ at Node 12, the robot performs ⟨swapping at PM_3, moving to LL (m_{30}), loading wafer W_1 to LL (l_0), unloading wafer W_5 from LL (u_0), moving to PM_1 (m_{01}), and waiting at PM_1 (ω_1^8)⟩.

For Case 2, the system evolves as shown in FIG. **3**C and its states are as follows:

To reach $M_{42}=\{W_3(1), W_1(4), W_2(3), R_2(\Theta)\}$ at Node 9, the robot performs ⟨moving to PM_2 (m_{32}), waiting at PM_2 (ω_2^4), unloading wafer W_2 from PM_2 (u_2), moving to PM_3 (m_{23}), waiting at PM_3 (ω_3^4), swapping at PM_3, moving to PM_2 (m_{32}), and loading wafer W_1 to PM_2 (l_2)⟩;

To reach $M_{52}=\{W_3(1), W_1(4), W_2(3), R_1(W_4(1))\}$ at Node 5F, the robot performs ⟨moving to LL (m_{20}), unloading wafer W_4 from LL (u_0), moving to PM_1 (m_{01}), and waiting at PM_1 (ω_1^5)⟩;

To reach $M_{62}=\{W_4(1), W_3(2), W_2(3), R_3(W_1(5))\}$ at Node 8F, the robot performs ⟨swapping at PM_1, moving to PM_2 (m_{12}), waiting at PM_2 (ω_2^6), swapping at PM_2, moving to PM_3 (m_{23}), and waiting at PM_3 (ω_3^6)⟩;

To reach $M_{72}=\{W_4(1), W_2(4), W_1(5), R_3(W_3(3))\}$ at Node 11F, the robot performs ⟨swapping at PM_3, moving to PM_2 (m_{32}), waiting at PM_2 (ω_2^7), swapping at PM_2, moving to PM_3 (m_{23}), and waiting at PM_3 (ω_3^7)⟩;

To reach $M_{82}=\{W_4(1), W_2(4), W_3(3), R_1(W_5(1))\}$ at Node 12, the robot performs ⟨swapping at PM_3, moving to LL (m_{30}), loading wafer W_1 to LL (l_0), unloading wafer W_5 from LL (u_0), moving to PM_1 (m_{01}), and waiting at PM_1 (ω_1^8)⟩.

Let Γ_k denote the time from an initial state at Node 1 to a terminal state at Node k. Thus, to obtain the shortest feasible path Γ_k, the key is to search a critical path in the PERT charts shown in FIGS. **3**B and **3**C. One adopts the following algorithm to calculate Γ_k.

Algorithm 1: If a dual-arm cluster tool with wafer revisiting operates as shown in FIG. **3**B from the initial state to the steady state, the minimal time Γ_{13} is computed as follow.

1) $\Gamma_2=\alpha+\mu+\beta$;
2) $\Gamma_{2F}=\Gamma_2+\max\{a_1, (2\mu+\alpha)\}$;
3) $\Gamma_3=\Gamma_{2F}+\lambda$;
4) $\Gamma_4=\Gamma_3+\mu+\beta$;
5) $\Gamma_{3F}=\max\{(\Gamma_3+a_1), (\Gamma_4+2\mu+\alpha)\}$;
6) $\Gamma_5=\Gamma_{3F}+\lambda$;
7) $\Gamma_{4F}=\max\{(\Gamma_4+a_2), (\Gamma_{3F}+\lambda+\mu)\}$;
8) $\Gamma_6=\Gamma_{4F}+\lambda$;
9) $\Gamma_7=\Gamma_6+\mu+\beta$;
10) $\Gamma_{7F}=\Gamma_7+a_3$;
11) $\Gamma_{6F}=\max\{(\Gamma_{7F}+\alpha+\mu), (\Gamma_6+a_2)\}$;
12) $\Gamma_8=\Gamma_{6F}+\lambda$;
13) $\Gamma_9=\Gamma_8+\mu+\beta$;
14) $\Gamma_{5F}=\max\{(\Gamma_9+2\mu+\alpha), (\Gamma_5+a_1)\}$;
15) $\Gamma_{8F}=\max\{(\Gamma_{5F}+\mu+\lambda), (\Gamma_8+a_2)\}$;
16) $\Gamma_{10}=\Gamma_{8F}+\lambda$;
17) $\Gamma_{9F}=\max\{(\Gamma_{10}+\mu), (\Gamma_9+a_3)\}$;
18) $\Gamma_{11}=\Gamma_{9F}+\lambda$;
19) $\Gamma_{10F}=\max\{(\Gamma_{11}+\mu), (\Gamma_{10}+a_2)\}$;
20) $\Gamma_{11F}=\max\{(\Gamma_{10F}+\lambda+\mu), (\Gamma_{11}+a_3)\}$;
21) $\Gamma_{12}=\max\{(\Gamma_{5F}+\lambda+a_1), (\Gamma_{11F}+\lambda+2\mu+\alpha+\beta)\}$;
22) $\Gamma_{13}=\Gamma_{12}+\lambda$;
23) Stop;

At the initial state, the cluster tool is empty. Then, one shows how Algorithm 1 is obtained. From the PERT chart in FIG. **3**B for Case 1, one knows:

To reach the state at Node 2 from Node 1 takes $\Gamma_2=\alpha+\mu+\beta$ time units at least.

To reach Node 2F from Node 2, there are two sequences of operations that must be finished. One takes a_1 time units to finish the process, and the other takes $2\mu+\alpha$ time units to finish transportation. Hence, the time needed to reach Node 2F is $\max\{a_1, (2\mu+\alpha)\}$ and Statement 2 holds.

To reach Node 3 takes at least $\Gamma_{2F}+\lambda$ time units and Statements 3 holds.

To reach Node 4 takes $\Gamma_3+\mu+\beta$ time units and Statement 4 hold.

To reach Node 3F, there are two sequences to complete. One takes Γ_3+a_1 time units to finish the process from Node 3, and the other takes $\Gamma_4+2\mu+\alpha$ time units to finish transportation. Hence, the time needed to reach Node 3F is $\max\{(\Gamma_3+a_1), (\Gamma_4+2\mu+\alpha)\}$ and Statement 5 holds.

To reach Node 5 takes at least $\Gamma_{3F}+\lambda$ time units. Thus, Statement 6 holds.

To reach Node 4F, it has two sequences to complete. One takes Γ_4+a_2 time units to finish the process from Node 4, and the other takes $\Gamma_{3F}+\lambda+\mu$ time units to finish transportation. Hence, the time needed to reach Node 4F is $\max\{(\Gamma_3+a_1), (\Gamma_4+2\mu+\alpha)\}$ and Statement 7 holds.

To reach Nodes 6, 7, and 7F takes at least $\Gamma_{4F}+\lambda$, $\Gamma_6+\mu+\beta$ and Γ_7+a_3 time units, respectively. Thus, Statements 8-10 hold.

To reach Node 6F, it has two sequences to complete. One takes $\Gamma_{7F}+\alpha+\mu$ time units to finish transportation, and the other takes Γ_6+a_2 time units to finish the process. Hence, the time needed to reach Node 6F is $\max\ \{(\Gamma_{7F}+\alpha+\mu), (\Gamma_6+a_2)\}$ and Statement 11 holds.

To reach Nodes 8 and 9 takes $\sigma_{6F}+\lambda$ and $\Gamma_8+\mu+\beta$ time units, respectively. Thus, Statements 12 and 13 hold.

To reach Node 5F, it has two sequences to complete. One takes $\Gamma_9+2\mu+\alpha$ time units to finish transportation, and the other takes Γ_5+a_1 time units to finish the process from Node 5. Hence, the time needed to reach Node 5F is $\max\{(\Gamma_9+2\mu+\alpha), (\Gamma_5+a_1)\}$ time units and Statement 14 holds.

To reach Node 8F, it has two sequences to complete. One takes $\Gamma_{5F}+\mu+\lambda$ time units to finish transportation, and the other takes Γ_8+a_2 time units to finish the process from Node 8. Hence, the time needed to reach Node 8F is $\max\ \{(\Gamma_{5F}+\mu+\lambda), (\Gamma_8+a_2)\}$ time units and Statement 15 holds.

To reach Node 10 takes at least $\Gamma_{8F}+\lambda$ time units and Statement 16 holds.

To reach Node 9F, it has two sequences to complete. One takes $\Gamma_{10}+\mu$ time units to finish transportation, and the other takes Γ_9+a_3 time units to finish the process. Hence, the time needed to reach Node 9F is $\max\{(\Gamma_{10}+\mu), (\Gamma_9+a_3)\}$ time units and Statement 17 holds.

To reach Node 11 takes at least $\Gamma_{9F}+\lambda$ time units and Statement 18 holds.

To reach Node 10F, it has two sequences to complete. One takes $\Gamma_{11}+\mu$ time units to finish transportation, and the other takes $\Gamma_{10}+a_2$ time units to finish the process from Node 10. Hence, the time needed to reach Node 10F is $\max\{(\Gamma_{11}+\mu), (\Gamma_{10}+a_2)\}$ time units and Statement 19 holds.

To reach Node 11F, it has two sequences to complete. One takes $\Gamma_{10F}+\lambda+\mu$ time units to finish transportation, and the other takes $\Gamma_{11}+a_3$ time units to finish the process from Node 11. Hence, the time from Nodes 11 to 11F is $\max\{(\Gamma_{10F}+\lambda+\mu), (\Gamma_{11}+a_3)\}$ time units and Statement 20 holds.

To reach Node 12, one sequence takes $(\Gamma_{11F}+\lambda+2\mu+\alpha+\beta)$ time units to finish transportation, and the other takes $(\Gamma_{5F}+\lambda+a_1)$ time units to finish the process. Hence, the time needed to reach Node 12 is $\max\{(\Gamma_{5F}+\lambda+a_1), (\Gamma_{11F}+\lambda+2\mu+\alpha+\beta)\}$ time units and Statement 21 holds.

To reach Node 13 takes $\Gamma_{12}+\lambda$ time units.

A critical path from Node 1 to Node 13 could be found by Algorithm 1. So for Case 1, the time Γ_{13} is shortest time by Algorithm 1 for a dual-cluster tool to reach the first desired steady state from the initial state.

Algorithm 2: If a dual-arm cluster tool with wafer revisiting operates as shown in FIG. **3**C from the initial state, the minimal time Γ_{13} could be obtained as follow.

```
1) $\Gamma_2=\alpha+\mu+\beta$;
2) $\Gamma_{2F}=\Gamma_2+\max\{a_1, (2\mu+\alpha)\}$;
3) $\Gamma_3=\Gamma_{2F}+\lambda$;
4) $\Gamma_4=\Gamma_3+\mu+\beta$;
5) $\Gamma_{3F}=\max\{(\Gamma_3+a_1), (\Gamma_4+2\mu+\alpha)\}$;
6) $\Gamma_5=\Gamma_{3F}+\lambda$;
7) $\Gamma_{4F}=\max\{(\Gamma_4+a_2), (\Gamma_{3F}+\lambda+\mu)\}$;
8) $\Gamma_6=\Gamma_{4F}+\lambda$;
9) $\Gamma_7=\Gamma_6+\mu+\beta$;
10) $\Gamma_{6F}=\max\ \{(\Gamma_6+a_2), (\Gamma_7+\mu)\}$;
11) $\Gamma_{7F}=\max\{(\Gamma_7+a_3), (\Gamma_{6F}+\mu+\alpha)\}$;
12) $\Gamma_8=\Gamma_{7F}+\lambda$;
13) $\Gamma_9=\Gamma_8+\mu+\beta$;
14) $\Gamma_{5F}=\max\{(\Gamma_5+a_1), (\Gamma_9+2\mu+\alpha)\}$;
15) $\Gamma_{9F}=\max\{(\Gamma_9+a_2), (\Gamma_{5F}+\mu+\lambda)\}$;
16) $\Gamma_{10}=\Gamma_{9F}+\lambda$;
17) $\Gamma_{8F}=\max\{(\Gamma_8+a_3), (\Gamma_{10}+\mu)\}$;
18) $\Gamma_{11}=\Gamma_{8F}+\lambda$;
19) $\Gamma_{10F}=\max\{(\Gamma_{10}+a_2), (\Gamma_{11}+\mu)\}$;
20) $\Gamma_{11F}=\max\{(\Gamma_{11}+a_3), (\Gamma_{10F}+\mu+\lambda)\}$;
21) $\Gamma_{12}=\max\{(\Gamma_{5F}+\lambda+a_1), (\Gamma_{11F}+\lambda+2\mu+\alpha+\beta)\}$;
22) $\Gamma_{13}=\Gamma_{12}+\lambda$;
23) Stop;
```

Similar to Algorithm 1, a critical path from Node 1 to Node 13 could be found by Algorithm 2 for Case 2. Thus, Γ_{13} obtained by Algorithm 2 represents the shortest time for the cluster tool to reach the first desired steady state at Node 13 from the initial state.

Up to now, one knows that the transient process obtained by Algorithms 1 and 2 from the initial state to the state at Node 13 is the shortest time for Cases 1 and 2, respectively. Then, if the cycle time of the cluster tool is a constant and optimal in the following evolutions, by simulation results of Algorithm 1 and 2, one can know that which scheduling method is better than traditional method which adopts the concept of virtual wafers [Wu et al., 2013b]. Observing the processes of M_{51} to M_{61} and M_{52} to M_{62}, M_{61} to M_{71} and M_{62} to M_{72}, and M_{71} to M_{81} and M_{72} to M_{82}, the robot activities for transforming the state from M_{i1} to $M_{(i+1)1}$ are same as the one for M_{i2} to $M_{(i+1)2}$, $i \in \{5, 6, 7\}$. Thus, one uses M_i to denote states M_{i1} and M_{i2}, $i \in \{5, 6, 7\}$. Then, after state M_8, the following evolutions of the cluster tool is shown.

From M_{8+3i} to $M_{9+3i}=\{W_{(5+i)}(1), W_{(4+i)}(2), W_{(3+i)}(3), R_3(W_{(2+i)}(5))\}$, $i \geq 0$, the robot performs a sequence of robot activities: $\langle$ swaps at PM_1, moves to PM_2 (m_{12}), waits at PM_2 (ω_2^{9+3i}), swaps at PM_2, moves to PM_3 (m_{23}), and waits at PM_3 (ω_3^{9+3i})$\rangle$;

From M_{9+3i} to $M_{10+3i}=\{W_{(5+i)}(1), W_{(3+i)}(4), W_{(2+i)}(5), R_3(W_{(4+i)}(3))\}$, $i \geq 0$, the robot performs a sequence of robot activities: $\langle$ swaps at PM_3, moves to PM_2 (m_{32}), waits at PM_2 (ω_2^{10+3i}), swaps at PM_2, moves to PM_3 (m_{23}), and waits at PM_3 (ω_3^{10+3i})$\rangle$;

From M_{10+3i} to $M_{11+3i}=\{W_{(5+i)}(1), W_{(3+i)}(4), W_{(4+i)}(3), R_1(W_{(6+i)}(1))\}$, $i \geq 0$, the robot performs a sequence of robot activities: $\langle$ swaps at PM_3, moves to LL (m_{30}), loads $W_{(2+i)}$ into LL (l_0), unloads $W_{(6+i)}$ from LL (u_0), moves to PM_1 (m_{01}), and waits at PM_1 (ω_1^{11+3i})$\rangle$.

When a wafer is processed, it is unloaded by the robot. It takes a_i time units at Step i to process the wafer and λ time units to swap, $i \in N_3$. Let $\Pi_i=a_i+\lambda$ denote the time needed for completing a wafer at Step i, and $\Pi_{local}=\max\{\Pi_2, \Pi_3\}$. Let ϕ_1 denote the local robot cycle time without considering the robot waiting time. Then $\phi_1=2\mu+2\lambda$. Let θ_{ij}, $j>i\geq 0$, denote the robot task sequence which transfers M_i to M_j and Π_{ij} denote the time taken by θ_{ij}. Then, one has the following theorem.

Theorem 1: For a dual-arm cluster tool with wafer revisiting with the start-up transient process scheduled by Algorithms 1 or 2, if $\Pi_1 \leq \Pi_{local}+\psi_1$, then $\omega_1^{8+3i}=0$ must hold, $i \geq 0$.

Proof: Case 1: The start-up transient process is scheduled by Algorithm 1. Observe the robot task sequence θ_1 during the period from M_5 to M_7, where $\theta_1=\langle$ swaps at PM_2, moves to PM_3 (m_{23}), waits at PM_3 (ω_3^6), swaps at PM_3, moves to PM_2 (m_{32}), waits at PM_2 (ω_2^7), swaps at PM_2, moves to PM_3 (m_{23}), and waits at PM_3 (ω_3^7)$\rangle$. The earliest strategy implies that the robot task time taken by θ_1 must be greater than $\Pi_2+\lambda+\mu$ or

$\Pi_3+\lambda+\mu$. Therefore, the time needed for θ_1 must be greater than $\Pi_{local}+\lambda+\mu$ time units. Thus, the robot task sequence $\theta_{58}=\langle$ swaps at PM_1, moves to PM_2 (m_{12}), waits at PM_2 (ω_2^{6}), robot task sequence θ_1, swaps at PM_3, moves to LL (m_{30}), loads W_1 into LL (l_0), unloads W_5 from LL (u_0), moves to PM_1 (m_{01}), and waits at PM_1 (ω_1^{8})$\rangle$ must take $\Pi_{local}+3\lambda+4\mu+\alpha+\beta=\Pi_{local}+\psi_1$ time units at least because θ_1 must take $\Pi_{local}+\lambda+\mu$ time units at least. Then, by the assumption of $\Pi_1\leq\Pi_{local}+\psi_1$, when the system reaches marking M_9 for swapping at p_1, $\omega_1^{8}=0$ must hold. Similarly, one can prove that $\omega_1^{8+3i}=0$, $i\geq0$.

Case 1: The start-up transient process is scheduled by Algorithm 2. Similar to Case 1, one can have $\omega_1^{8+3i}=0$, $i\geq0$.

Hence, the conclusion holds.

From Theorem 1, it shows that the time needed for the robot task sequence θ_1 must be greater than $\Pi_{local}+\lambda+\mu$ time units. θ_1 consists of a local cycle, a swapping operation, and a robot moving activity. Thus, one has that the local cycle takes Π_{local} time units at least. With the start-up transient process scheduled by Algorithms 1 or 2, when the system reaches marking M_{8+3i}, the robot would swap at PM_1 for unloading a wafer which is completed in PM_1. Notice that this wafer has stayed in PM_1 for a whole period including a global cycle and a local one. Because the global cycle without the robot waiting time being considered takes ψ_1 time units, this wafer must stay in PM_1 for $\Pi_{local}+\psi_1$ time units at least. With the assumption of $\Pi_1\leq\Pi_{local}+\psi_1$, one can easily know that, when the robot arrives at PM_1 for unloading this wafer, it must be completed. Therefore, no waiting is needed before swapping at PM_1. Then, based on Theorem 1, one can have the following theorem.

Theorem 2: For a dual-arm cluster tool handling an ALD process, for the following operations of the system after the robot performs all the tasks before Node 13 shown in FIGS. **3**B or **3**C, if $\Pi_1\leq\Pi_{local}+\psi_1$, then the cycle time of the system is a constant, i.e.,

$$\Pi_{(8+3i)(11+3i)}=\begin{cases}2\Pi_{local},\text{ if }\ \Pi_{local}\geq\psi_1\\ \Pi_{local}+\psi_1,\text{ if }\ \psi_1>\Pi_{local}\geq\varphi_1\\ \psi_1+\varphi_1,\text{ if }\ \varphi_1>\Pi_{local}\end{cases}\tag{1}$$

Proof: One has the following five situations. Situation 1: $\Pi_3\geq\Pi_2$ and $\Pi_3\geq\psi_1$. From the robot task sequence $\theta_{(6+3i)(9+3i)}=\langle$ swaps at PM_3, moves to PM_2 (m_{32}), waits at PM_2 (ω_2^{7+3i}), swaps at PM_2, moves to PM_3 (m_{23}), waits at PM_3 (ω_3^{7+3i}), swaps at PM_3, moves to LL (m_{30}), loads a completed wafer into LL (l_0), unloads a row wafer from LL (u_0), moves to PM_1 (m_{01}), waits at PM_1 (ω_1^{8+3i}), swaps at PM_1, moves to PM_2 (m_{12}), waits at PM_2 (ω_2^{9+3i}), swaps at PM_2, moves to PM_3 (m_{23}), and waits at PM_3 (ω_3^{9+3i}))$\rangle$, $i\geq0$, one has $\omega_3^{7+3i}\leq\Pi_3-\phi_1$ because $\Pi_3\geq\psi_1>\phi_1$. In $\theta_{(6+3i)(9+3i)}$, $i\geq0$, for the robot task sequence $\theta_2=\langle$ swaps at PM_2, moves to PM_3 (m_{23}), waits at PM_3 (ω_3^{7+3i}), swaps at PM_3, moves to LL (m_{30}), loads a completed wafer into LL (l_0), unloads a raw wafer from LL (u_0), moves to PM_1 (m_{01}), waits at PM_1 (ω_1^{8+3i}), swaps at PM_1, moves to PM_2 (m_{12}), waits at PM_2 (ω_2^{9+3i}))$\rangle$, it takes $\psi_1+\omega_3^{7+3i}+\omega_2^{9+3i}$ time units. Thus, if $\psi_1+\omega_3^{7+3i}\geq\Pi_2$, one has $\omega_2^{9+3i}=0$, and otherwise, $\omega_2^{9+3i}=\Pi_2-(\psi_1+\omega_3^{7+3i})$. From $\theta_{(7+3i)(9+3i)}=\langle$ swaps at PM_3, moves to LL (m_{30}), loads a completed wafer into LL (l_0), unloads a raw wafer from LL (u_0), moves to PM_1 (m_{01}), waits at PM_1 (ω_1^{8+3i}), swaps at PM_1, moves to PM_2 (m_{12}), waits at PM_2 (ω_2^{9+3i}), swaps at PM_2, moves to PM_3 (m_{23}), and waits at PM_3 (ω_3^{9+3i}))$\rangle$, $i\geq0$, if $\omega_2^{9+3i}=0$, one has $\omega_2^{9+3i}+\omega_3^{9+3i}=\Pi_3-\psi_1$ because $\Pi_3\geq\psi_1$, and if $\omega_2^{9+3i}=\Pi_2-(\psi_1+\omega_3^{7+3i})>0$, one also has $\omega_2^{9+3i}+\omega_3^{9+3i}=\Pi_3-\psi_1$ because $0<\omega_2^{9+3i}=\Pi_2-(\psi_1+\omega_3^{7+3i})\leq\Pi_2-\psi_1\leq\Pi_3-\psi_1$. From $\theta_{(8+3i)(10+3i)}=\langle$ swaps at PM_1, moves to PM_2 (m_{12}), waits at PM_2 (ω_2^{9+3i}), swaps at PM_2, moves to PM_3 (m_{23}), waits at PM_3 (ω_3^{9+3i}), swaps at PM_3, moves to PM_2 (m_{32}), waits at PM_2 (ω_2^{10+3i}), swaps at PM_2, moves to PM_3 (m_{23}), and waits at PM_3 (ω_3^{10+3i}))$\rangle$, $i\geq0$, one has $\omega_2^{10+3i}=\max\{\Pi_2-\phi_1-\omega_3^{9+3i}, 0\}$. If $\omega_2^{10+3i}=0$, one has $\omega_2^{10+3i}+\omega_3^{10+3i}=\Pi_3-\phi_1$. If $\omega_2^{10+3i}=\Pi_2-\phi_1-\omega_3^{9+3i}$, one also has $\omega_2^{10+3i}+\omega_3^{10+3i}=\Pi_3-\phi_1$ because $\omega_2^{10+3i}=\Pi_2-\phi_1-\omega_3^{9+3i}\leq\Pi_2-\phi_1\leq\Pi_3-\phi_1$. Thus, the robot task sequence $\theta_{(8+3i)(11+3i)}$, $i\geq0$ takes $\psi_1+\phi_1+\omega_2^{9+3i}+\omega_3^{9+3i}+\omega_2^{10+3i}+\omega_3^{10+3i}=\psi_1+\phi_1+\Pi_3-\psi_1+\Pi_3-\phi_1=2\Pi_3$ time units.

Situation 2: $\Pi_2>\Pi_3$ and $\Pi_2\geq\psi_1$. Similar to Situation 1, one has $\omega_2^{9+3i}+\omega_3^{9+3i}=\Pi_2-\psi_1$ and $\omega_2^{10+3i}+\omega_3^{10+3i}=\Pi_2-\phi_1$. Thus, the robot task sequence $\theta_{(8+3i)(11+3i)}$, $i\geq0$, takes $\psi_1+\phi_1+\omega_2^{9+3i}+\omega_3^{9+3i}+\omega_2^{10+3i}+\omega_3^{10+3i}=2\Pi_2$ time units.

Situation 3: $\Pi_3\geq\Pi_2$ and $\psi_1>\Pi_3\geq\phi_1$. Similar to Situation 1, one has $\omega_2^{9+3i}+\omega_3^{9+3i}=0$ and $\omega_2^{10+3i}+\omega_3^{10+3i}=\Pi_3-\phi_1$. Thus, the robot task sequence $\theta_{(8+3i)(11+3i)}$, $i\geq0$, takes $\psi_1+\phi_1+\omega_2^{9+3i}+\omega_3^{9+3i}+\omega_2^{10+3i}+\omega_3^{10+3i}=\Pi_3+\psi_1$ time units.

Situation 4: $\Pi_2>\Pi_3$ and $\psi_1>\Pi_2\geq\phi_1$. Similar to Situation 1, one has $\omega_2^{9+3i}+\omega_3^{9+3i}=0$ and $\omega_2^{10+3i}+\omega_3^{10+3i}=\Pi_2-\phi_1$. Thus, the robot task sequence $\theta_{(8+3i)(11+3i)}$, $i\geq0$, takes $\psi_1+\phi_1+\omega_2^{9+3i}+\omega_3^{9+3i}+\omega_2^{10+3i}+\omega_3^{10+3i}=\Pi_2+\psi_1$ time units.

Situation 5: $\phi_1>\Pi_{local}$. One can easily have that $\omega_2^{9+3i}+\omega_3^{9+3i}=0$ and $\omega_2^{10+3i}+\omega_3^{10+3i}=0$. Thus, the robot task sequence $\theta_{(8+3i)(11+3i)}$, $i\geq0$, takes $\psi_1+\phi_1+\omega_2^{9+3i}+\omega_3^{9+3i}+\omega_2^{10+3i}+\omega_3^{10+3i}=\psi_1+\phi_1$ time units.

Therefore, the conclusion holds.

By Theorem 2, if $\Pi_{local}\geq\psi_1$ or $\Pi_{local}\geq\phi_1$, the system operates in the process-bound region for the local cycle. Thus, the local cycle takes Π_{local} time units. If $\Pi_{local}<\phi_1$, the system operates in the transport-bound region for the local cycle. For this situation, the local cycle takes ϕ_1 time units. Then, if $\Pi_{local}\geq\psi_1$, the system operates in the process-bound for the global cycle which takes Π_{local} time units. If $\Pi_{local}<\psi_1$, the system operates in the transport-bound for the global cycle which takes ψ_1 time units. Therefore, expression (1) can hold. The following theorem is for another case.

Theorem 3: For a dual-arm cluster tool handling an ALD process, for the following operations of the system after the robot performs all the tasks before Node 12 (except robot waiting ω_1^{8}) shown in FIGS. **3**B or **3**C, if $2\Pi_{local}\geq\Pi_1>\Pi_{local}\psi_1$ and $\Pi_{local}\geq\psi_1$, then the cycle time of the system is $2\Pi_{local}$ time units.

Proof: One has two situations only. Situation 1: $\Pi_3\geq\Pi_2$. It follows from Situation 1 in Theorem 2 that one has $\omega_3^{7+3i}\leq\Pi_3-\phi_1$. Then, in $\theta_{(6+3i)(9+3i)}$, $i\geq0$, for the robot task sequence $\theta_3=\langle$ swaps at PM_2, moves to PM_3 (m_{23}), waits at PM_3 (ω_3^{7+3i}), swaps at PM_3, moves to LL (m_{30}), loads a completed wafer into LL (l_0), unloads a raw wafer from LL (u_0), moves to PM_1 (m_{01}), waits at PM_1 (ω_1^{8+3i}), swaps at PM_1, moves to PM_2 (m_{12}), waits at PM_2 (ω_2^{9+3i}))$\rangle$, it takes $\psi_1+\omega_3^{7+3i}+\omega_2^{9+3i}+\omega_1^{8+3i}$ time units. Thus if, $\psi_1+\omega_3^{7+3i}+\omega_1^{8+3i}\geq\Pi_2$, one has $\omega_2^{9+3i}=0$, otherwise ω_2^{9+3i}**32** $\Pi_2-(\psi_1+\omega_3^{7+3i}+\omega_1^{8+3i})$. Observing $\theta_{(7+3i)(9+3i)}=\langle$ swaps at PM_3, moves to LL (m_{30}), loads a completed wafer into LL (l_0), unloads a raw wafer from LL (u_0), moves to PM_1 (m_{01}), waits at PM_1 (ω_1^{8+3i}), swaps at PM_1, moves to PM_2 (m_{12}), waits at PM_2 (ω_2^{9+3i}), swaps at PM_2, moves to PM_3 (m_{23}), and waits at PM_3 (ω_3^{9+3i}))$\rangle$, $i\geq0$, if $\omega_2^{9+3i}=0$, one has $\omega_1^{8+3i}+\omega_2^{9+3i}+\omega_3^{9+3i}=\Pi_3-\psi_1$ because of $\omega_1^{8+3i}\leq\Pi_1-(\Pi_{local}+\psi_1)$ and $2\Pi_{local}\geq\Pi_1$ leading to $\omega_1^{8+3i}+\psi_1\leq\Pi_1-(\Pi_3+\psi_1)+\psi_1=\Pi_1-\Pi_3\leq\Pi_3$. If $\omega_2^{9+3i}=\Pi_2-(\psi_1+\omega_3^{7+3i}+\omega_1^{8+3i})>0$, one also has $\omega_1^{8+3i}+\omega_2^{9+3i}+\omega_3^{9+3i}=\Pi_3-\psi_1$ because of $0<\omega_2^{9+3i}=\Pi_2-(\psi_1+\omega_3^{7+3i}+\omega_1^{8+3i})\leq\Pi_2-\psi_1\leq\Pi_3-\psi_1$. Then, based on Situation 1 in Theorem 2, one has $\omega_2^{10+3i}+\omega_3^{10+3i}=\Pi_3-\phi_1$.

Observing the robot task sequence $\theta_{(7+3i)(11+3i)}$, $i\geq 0$, one sets a robot task sequence $\sigma_{8+3i}=\langle$ waits at PM_1 (ω_1^{8+3i}), swaps at PM_1, moves to PM_2 (m_{12}), waits at PM_2 (ω_2^{9+3i}), swaps at PM_2, moves to PM_3 (m_{23}), waits at PM_3 (ω_3^{9+3i}), swaps at PM_3, moves to PM_2 (m_{32}), waits at PM_2 (ω_2^{10+3i}), swaps at PM_2, moves to PM_3 (m_{23}), waits at PM_3 (ω_3^{10+3i}), swaps at PM_3, moves to LL (m_{30}), loads a completed wafer into LL (l_0), unloads a raw wafer from LL (u_0), and moves to PM_1 (m_{01})$\rangle$, $i\geq 0$. Thus, the robot task sequence σ_{8+3i} takes $\psi_1+\phi_1+\omega_1^{8+3i}+\omega_2^{9+3i}+\omega_3^{9+3i}+\omega_2^{10+3i}+\omega_3^{10+3i}=\psi_1+\phi_1+\Pi_3-\psi_1+\Pi_3-\phi_1=2\Pi_3$ time units.

Situation 2: $\Pi_2>\Pi_3$. Similar to Situation 1, one has $\omega_1^{8+3i}+\omega_2^{9+3i}+\omega_3^{9+3i}=\Pi_2-\psi_1$ and $\omega_2^{10+3i}+\omega_3^{10+3i}=\Pi_2-\phi_1$. Thus, the robot task sequence σ_{8+3i}, $i\geq 0$, takes $\psi_1+\phi_1+\omega_1^{8+3i}+\omega_2^{9+3i}+\omega_3^{9+3i}+\omega_2^{10+3i}+\omega_3^{10+3i}=2\Pi_3$ time units.

Therefore, the conclusion holds.

For this case, since $\Pi_{local}\geq\psi_1$, one has that the system operates in the process-bound region for both the local and global cycles. With the assumption that $2\Pi_{local}\geq\Pi_1$, the local and global cycles take $2\Pi_{local}$ time units. However, if $\Pi_1>2\Pi_{local}$ and $\Pi_{local}\geq\psi_1$, one has the following result.

Theorem 4: For a dual-arm cluster tool for an ALD process, for the following operations of the system after the robot performs all the tasks before Node 13 shown in FIGS. **3**B or **3**C, if $\Pi_1>2\Pi_{local}$ and $\Pi_{local}\geq\psi_1$, then the cycle time of the system is Π_1 time units.

Proof: It is assumed that a robot task sequence $\theta_{(8+3i)(11+3i)}$, $i\geq 0$, takes η_1 time units and $\eta_1<\Pi_1$ holds. However, during the period of M_{8+3i} to M_{11+3i}, $i\geq 0$, there is a wafer staying at PM_1. After performing $\theta_{(8+3i)(11+3i)}$, the robot needs to swap at PM_1 for unloading the wafer. With the assumption of $\eta_1<\Pi_1$, if the robot unloads the wafer, this wafer must be not completed. Therefore, one has $\eta_1\geq\Pi_1$.

It is assumed that there exists a robot task sequence $\theta_{(8+3i)(11+3i)}$ which takes η_1 time units and $\eta_1>\Pi_1$ holds. With the earliest strategy, $\omega_1^{8+3i}=0$ because that, if $\omega_1^{8+3i}>0$, the wafer staying at PM_1 during $\theta_{(8+3i)(11+3i)}$ must just be completed such that $\theta_{(8+3i)(11+3i)}$ takes Π_1 time units leading to a conflict with the assumption of $\eta_1>\Pi_1$. Then, there are two situations only.

Situation 1: $\Pi_3\geq\Pi_2$. In $\theta_{(8+3i)(10+3i)}$, there is a robot task sequence $\theta_4=\langle$ swaps at PM_2, moves to PM_3 (m_{23}), waits at PM_3 (ω_3^{9+3i}), swaps at PM_3, moves to PM_2 (m_{32}), waits at PM_2 (ω_2^{10+3i}), swaps at PM_2, moves to PM_3 (m_{23}), and waits at PM_3 (ω_3^{10+3i})$\rangle$. If $\omega_3^{9+3i}=0$, with the earliest strategy, one has $\omega_2^{10+3i}=\max\{\Pi_2-\phi_1, 0\}$ holds. This means that $0\leq\omega_3^{10+3i}=\Pi_3-\phi_1-\omega_2^{10+3i}$ holds because of $\Pi_3-\phi_1\geq\max\{\Pi_2-\phi_1, 0\}$. Therefore, θ_4 takes $\Pi_3+\lambda+\mu+\omega_3^{9+3i}$ time units, where $\omega_3^{9+3i}=0$. If $\omega_3^{9+3i}>0$, with the earliest strategy, one has $\omega_2^{10+3i}<\max\{\Pi_2-\phi_1, 0\}$ holds. This means that $0\leq\omega_3^{10+3i}=\Pi_3-\phi_1-\omega_2^{10+3i}$ holds because of $\Pi_3-\phi_1\geq\max\{\Pi_2-\phi_1, 0\}$. Therefore, θ_4 takes $\Pi_3+\lambda+\mu+\omega_3^{9+3i}$ time units, where $\omega_3^{9+3i}>0$. Thus, one has that the robot task sequence $\theta_{(8+3i)(11+3i)}$ takes $\eta_1=5\lambda+6\mu+\alpha+\beta+\omega_2^{9+3i}+\omega_3^{9+3i}+\omega_2^{10+3i}+\omega_3^{10+3i}+\omega_1^{11+3i}=\Pi_3+\psi_1+\omega_2^{9+3i}+\omega_3^{9+3i}$ time units. With the assumption of $\eta_1>\Pi_1$, one has $\omega_2^{9+3i}+\omega_3^{9+3i}>0$ because of $\Pi_3+\psi_1<\Pi_1$. If $\omega_3^{9+3i}>0$, with the earliest strategy and the robot task sequence $\theta_{(7+3i)(9+3i)}$, it means that $\omega_2^{9+3i}+\omega_3^{9+3i}\leq\Pi_3-\psi_1$. This makes that $\eta_1=\Pi_3+\psi_1+\omega_2^{9+3i}+\omega_3^{9+3i}\leq 2\Pi_3<\Pi_1$ which leads a conflict with the assumption of $\eta_1>\Pi_1$. If $\omega_3^{9+3i}=0$ and $\omega_2^{9+3i}>0$, with the earliest strategy and the robot task sequence $\theta_{(6+3i)(9+3i)}$, it means that $\omega_2^{9+3i}\leq\Pi_2-\psi_1$ and $\Pi_2>\psi_1$ must hold. At this time, $\eta_1=\Pi_3+\psi_1+\omega_2^{9+3i}+\omega_3^{9+3i}\leq\Pi_3+\Pi_2\leq 2\Pi_3<\Pi_1$ which leads a conflict with the assumption of $\eta_1>\Pi_1$. Therefore, one has $\eta_1\leq\Pi_1$ should hold. From above analysis, one knows that $\eta_1\geq\Pi_1$ should hold. Thus, $\eta_1=\Pi_1$ holds.

Situation 2: $\Pi_3<\Pi_2$. Similar to Situation 1, one can easily have $\eta_1=\Pi_1$.

Hence, the conclusion holds.

It is known that the wafer in PM_1 should stay for a whole period including a local cycle and a global one. Since $\Pi_1>2\Pi_{local}$, the wafer in PM_1 should stay for Π_1 time units at least. Therefore, the whole period takes Π_1 time units, or Theorem 4 holds. When $\Pi_1>\Pi_{local}+\psi_1$ and $\Pi_{local}<\psi_1$, one has the following theorem.

Theorem 5: For a dual-arm cluster tool handling an ALD process, for the following operations of the system after the robot performs all the tasks before Node 13 shown in FIG. **3**B or **3**C, if $\Pi_1>\Pi_{local}+\psi_1$ and $\phi_1\leq\Pi_{local}<\psi_1$, then the cycle time of the system is Π_1 time units.

Proof: By $\Pi_{local}<\psi_1$, one can easily have $\omega_2^{9+3i}+\omega_3^{9+3i}=0$, $i\geq 0$. Then, based on theorems 1-3, one has $\omega_2^{10+3i}+\omega_3^{10+3i}=\Pi_{local}-\phi_1$, $i\geq 0$. Thus, for the robot task sequence $\theta_{(8+3i)(11+3i)}$, $i\geq 0$, it takes $\psi_1+\phi_1+\omega_2^{9+3i}\omega_3^{9+3i}+\omega_2^{10+3i}+\omega_3^{10+3i}\omega_1^{11+3i}=\psi_1+\phi_1+\Pi_{local}-\phi_1+\omega_1^{11+3i}=\psi_1+\Pi_{local}+\omega_1^{11+3i}$. Because of $\Pi_1>\Pi_{local}+\psi_1$, one has $\omega_1^{11+3i}=\Pi_1-(\Pi_{local}+\psi_1)$ time units. This implies that the robot task sequence $\theta_{(8+3i)(11+3i)}$ takes Π_1 time units. Therefore, the theorem holds.

In this case, the local cycle takes Π_{local} time units. With $\Pi_{local}<\psi_1$, the global cycle takes ψ_1 time units at least. Due to $\Pi_1>\Pi_{local}+\psi_1$, one has that the whole period including a local and a global cycle must take Π_1 time units. Hence, Theorem 5 holds. When $\Pi_1>\Pi_{local}+\psi_1$ and $\Pi_{local}<\phi_1$, there are two cases. Case 1: $\Pi_1>\phi_1+\psi_1$; Case 2: $\Pi_1\leq\phi_1+\psi_1$. For these two cases, based on Theorem 5, one can easily have the following corollary.

Corollary 1: For a dual-arm cluster tool handling an ALD process, for the following operations of the system after the robot performs all the tasks before Node 13 shown in FIGS. **3**B or **3**C, if $\Pi_1>\Pi_{local}+\psi_1$ and $\Pi_{local}<\phi_1$, then the cycle time of the system is

$$\Pi_{(8+3i)(11+3i)}=\begin{cases}\Pi_1,\ \text{if}\ \ \Pi_1>\varphi_1+\psi_1\\ \varphi_1+\psi_1,\ \text{if}\ \ \Pi_1\leq\varphi_1+\psi_1\end{cases}\tag{2}$$

From Theorems 2, 4, and 5 and Corollary 1, one knows that, for a dual-arm cluster tool handling an ALD process, for the following operations of the system after the robot performs all the tasks before Node 13 shown in FIG. **3**B or **3**C, the system operates in a steady state and its cycle time is a constant. Thus, with Γ_{13} computed via Algorithms 1 and 2, one can know which scheduling strategy is better. Similarly, from Theorem 3, one can decide which scheduling strategy given by Algorithms 1 and 2 is better. Notice that both algorithms consisting of some analytical expressions. Therefore, it is very efficient.

For the ALD case, when a wafer revisits a step it requires exactly the same processing environment as it visits in its prior step. Thus, there is only one PM for each step. The results obtained in this present invention are based on the wafer flow pattern (PM_1, $(PM_2, PM_3)^2$). However, the proposed method can be applied to cases of parallel PMs with some adjustments. First, one need find a one-wafer cyclic schedule to schedule the tool. As it is pointed out in [Qiao et al., 2013], at each state, the process progress of wafers is different. Hence, for such cases, one can adjust these states in a similar way as ones for the wafer flow pattern (PM_1, $(PM_2, PM_3)^2$) such that this requirement can be met when there are parallel PMs. After a one-wafer schedule is obtained, the algorithms similar to Algorithms 1 and 2 can be developed. In this way, the results in the present invention can be extended to cases including parallel PMs in a step.

C. ILLUMINATIVE EXAMPLES

Example 1

The wafer processing time at Steps 1-3 is a_1=80 s, a_2=35 s, and a_3=85 s, respectively, loading and unloading time from LL is α=β=4 s, moving time from one PM to another is μ=3 s, and swapping time is λ=8 s.

If one adopts the existing scheduling method [Qiao et al., 2013], one knows the time to reach the steady state is 586 s by simulation. However, if $a_2 \leq 115$ s, one finds that the result obtained by Algorithm 1 is better than that obtained by Algorithm 2 and its use leads to 491 s to reach the steady state. This is 16.2% reduction. If $a_2 > 115$ s, the result obtained by Algorithm 2 is better than that obtained by Algorithm 1 and its use yields 612 s while the existing one needs 759 s. The time to reach the steady state is 612 s. This is 19.4% reduction. One can vary certain process time and examine how it impacts the time to reach the first desired steady state. Let A, B and C denote the time to reach the first desired steady state by Algorithm 1, Algorithm 2 and existing method, respectively. The horizontal and vertical axes denote the number of wafer and steady state time, respectively. The line of rhombus, circle, square and triangle denote the time a_2, A (Algorithm 1), B (Algorithm 2) and C (existing method), respectively, as shown in FIG. 4A. If processing time at PM_2 is small, Algorithm 1 should be used and after it increases to a certain value, i.e., 115 s. Algorithm 2 performs better. The detailed performance comparison results are shown in Table 4.1.

TABLE 4.1

for Example 1

Wafer	a_2	A	B	C	Reduction (%)
1	35	483	608	586	17.6
2	45	483	608	586	17.6
3	77	483	608	586	17.6
4	84	489	608	586	16.6
5	85	491	608	586	16.2
6	86	494	608	590	16.3
7	87	497	608	597	16.8
8	90	506	608	610	17.0
9	91	509	608	615	17.2
10	92	512	608	621	17.6
11	93	515	608	627	17.9
12	94	518	608	633	18.2
13	95	521	608	639	18.5
14	102	549	608	681	19.4
15	103	553	608	687	19.5
16	104	557	608	693	19.6
17	105	561	608	699	19.7
18	110	591	608	729	18.9
19	111	597	608	735	18.8
20	112	603	609	741	18.6
21	115	621	612	759	19.4
22	125	681	632	819	22.8
23	130	711	642	849	24.4
24	140	771	672	909	26.1
25	150	831	702	969	27.6

Example 2

The wafer processing time at Steps 1-3 is a_1=80 s, a_2=85 s, and a_3=35 s, respectively, loading and unloading time from LL is α=β=4 s, moving time from one PM to another is μ=3 s, and swapping time is λ=8 s.

According to [Qiao et al., 2013], the time to reach the steady state is 586 s by simulation. If $a_2 \leq 90$ s, one find that the result obtained by Algorithm 1 is better than that obtained by Algorithm 2 and its use leads to 491 s to reach the steady state. This is 16.2% reduction. If $a_2 > 90$ s, the result obtained by Algorithm 2 is better than that obtained by Algorithm 1 and its use yields 512 s while the existing one needs 635 s. The time to reach the steady state is 512 s. This is 19.4% reduction. By varying certain process time one can examine how it impacts the time to reach the desired first steady state. As shown in FIG. 4B and Table 4.2, the difference between the existing one and the proposed two algorithms is significant while that of the latter two is insignificant. When PM3's process time is small, Algorithm 1 is slightly better than Algorithm 2 and otherwise, Algorithm 2 is slightly preferred. The detailed performance comparison results are shown in Table 4.2.

TABLE 4.2

Example 2

Wafer	a_3	A	B	C	Reduction (%)
1	35	481	488	586	17.9
2	50	481	488	586	17.9
3	70	481	488	586	17.9
4	75	481	488	586	17.9
5	76	481	488	586	17.9
6	77	483	488	586	17.6
7	85	491	492	586	16.2
8	86	493	494	590	16.4
9	87	495	496	595	16.8
10	88	497	498	600	17.2
11	90	501	502	610	17.9
12	95	514	512	635	19.4
13	100	529	522	660	20.9
14	103	538	531	676	21.4
15	104	541	534	682	21.7
16	105	544	537	688	21.9
17	110	559	552	718	23.1
18	120	589	582	778	25.2
19	125	604	597	808	26.1
20	130	619	612	838	27.0
21	135	634	627	868	27.8
22	140	649	642	898	28.5
23	150	679	672	958	29.9

The industrial trend toward smaller wafer lot size and more variety of wafers tends to lead more and more frequent transient periods of cluster tools. Thus the research on transient processes becomes more important. Because a cluster tool has no buffer, scheduling and control become complex. Revisiting processes in wafer fabrication processes, e.g., an atomic layer deposition process, make such scheduling even more complex. Most previous researches focused on the steady state scheduling. Optimizing a schedule for start-up transient processes for dual-arm cluster tools is seriously lacking. The present invention presents the algorithms of the shortest time from an initial state to the first desired steady state and proves their effectiveness. Very significant reduction in time to reach the steady state is obtained in comparison with the existing method. Furthermore, when processing time fluctuates, one can observe the evolution of a transient process and how the desired steady state is reached, therefore enabling engineers to focus some critical processes for further improvement of the entire wafer fabrication process.

The embodiments disclosed herein may be implemented using general purpose or specialized computing devices, computer processors, or electronic circuitries including but not limited to digital signal processors (DSP), application specific integrated circuits (ASIC), field programmable gate arrays (FPGA), and other programmable logic devices configured or programmed according to the teachings of the present disclosure. Computer instructions or software codes running in the general purpose or specialized computing devices, computer processors, or programmable logic devices can readily be prepared by practitioners skilled in the software or electronic art based on the teachings of the present disclosure.

In some embodiments, the present invention includes computer storage media having computer instructions or software codes stored therein which can be used to program computers or microprocessors to perform any of the processes of the present invention. The storage media can include, but is not limited to, floppy disks, optical discs, Blu-ray Disc, DVD, CD-ROMs, and magneto-optical disks, ROMs, RAMs, flash memory devices, or any type of media or devices suitable for storing instructions, codes, and/or data.

The present invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The present embodiment is therefore to be considered in all respects as illustrative and not restrictive. The scope of the invention is indicated by the appended claims rather than by the foregoing description, and all changes that come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.

What is claimed is:

1. A computer implemented method for scheduling a dual-armed cluster tool with wafer revisiting for a transient process of a wafer fabrication system, the dual-armed cluster tool comprising a robot having two arms, and three process modules PMs, each for performing a wafer-processing Step, where PM_i is used for performing a processing Step i of the three wafer-processing steps for each wafer, $i \in N_3=\{1, 2, 3\}$, the wafer revisiting comprising a wafer flow pattern as $(PM_1, (PM_2, PM_3)^2)$ with $(PM_2, PM_3)^2$ being a 2-revisiting process, the method comprising:

obtaining, by a processor, a wafer processing time a_i for each of the three processing Step $i \in N_3=\{1, 2, 3\}$, a wafer loading time β for the robot, a wafer unloading time α for the robot, a moving time μ for the robot moving from one process module to another, and a swapping time λ for a swap operation executed by the robot as follows:
holding a wafer in one arm;
unloading a processed wafer from PM_i by the other arm;
rotating; and
loading the wafer into PM_i;

determining, by a processor, a plurality of algorithms of the transient process based on a plurality of robot task sequences θs for the robot, a plurality of system states M_j, j being a positive integer, and a program evaluation and review technique, wherein the transient process starts from an initial state among the system states and reaches to a steady state among the system states, and the initial state represents that no wafer is processed in the three PMs and the arms of the robot are empty, and the final states represents that a 3rd, a 1st and a 2nd wafers are processed in PM_1, PM_2, and PM_3 for their 1st, 4th, and 3rd operations respectively, wherein each of the system states represents a Node k, k being a positive integer, in the program evaluation and review technique, wherein the algorithms of the transient process execute a plurality of if-then instructions as follows:

if $\Pi_1 \le \Pi_{local}+\psi_1$, then a cycle time of the wafer fabrication system is a constant, and

$$\Pi_{(8+3i)(11+3i)} = \begin{cases} 2\Pi_{local}, \text{ if } \Pi_{local} \ge \psi_1 \\ \Pi_{local}+\psi_1, \text{ if } \psi_1 > \Pi_{local} \ge \varphi_1 \\ \psi_1+\varphi_1, \text{ if } \varphi_1 > \Pi_{local} \end{cases};$$

if $\Pi_1 > 2\Pi_{local}$ and $\Pi_{local} \ge \psi_1$, then a cycle time of the wafer fabrication system is Π_1 time units; and

if $\Pi_1 > \Pi_{local}+\psi_1$ and $\phi_1 \le \Pi_{local} < \psi_1$, then a cycle time of the wafer fabrication system is Π_1 time units;

where $\Pi_i=a_i+\lambda$ denote a time needed for completing a wafer at the processing Step i;

$\Pi_{local}=\max\{\Pi_2, \Pi_3\}$;

$\psi_1=3\lambda+4\mu+\alpha+\beta$;

Π_{ij} denotes a time taken by θ_{ij}, where θ_{ij}, $j \ge 0$, denotes a robot task sequence which transfers M_i to M_j; and

$\phi_1=2\mu+2\lambda$ denotes a local robot cycle time;

computing, by a processor, a time to reach the steady state from the initial state via performing each of the algorithms of the transient process based on the wafer processing time, the wafer loading time, the wafer unloading time, the moving time, and the swapping time; and

determining, by a processor, a corresponding robot task sequences given by the one of the algorithms of the transient process having a minimal time to reach the steady state among the computed times, wherein the determined robot task sequences are used for scheduling the transient process.

2. The method of claim **1**, wherein the time to reach the steady sate from the initial state is computed by one of the algorithms of the transient process as follows:

a. $\Gamma_2=\alpha+\mu+\beta$;
b. $\Gamma_{2F}=\Gamma_2+\max\{a_1, (2\mu+\alpha)\}$;
c. $\Gamma_3=\Gamma_{2F}+\lambda$;
d. $\Gamma_4=\Gamma_3+\mu+\beta$;
e. $\Gamma_{3F}=\max\{(\Gamma_3+a_1), (\Gamma_4+2\mu+\alpha)\}$;
f. $\Gamma_5=\Gamma_{3F}+\lambda$;
g. $\Gamma_{4F}=\max\{(\Gamma_4+a_2), (\Gamma_{3F}+\lambda+\mu)\}$;
h. $\Gamma_6=\Gamma_{4F}+\lambda$;
i. $\Gamma_7=\Gamma_6+\mu+\beta$;
j. $\Gamma_{7F}=\Gamma_7+a_3$;
k. $\Gamma_{6F}=\max\{(\Gamma_{7F}+\alpha+\mu), (\Gamma_6+a_2)\}$;
l. $\Gamma_8=\Gamma_{6F}+\lambda$;
m. $\Gamma_9=\Gamma_8+\mu+\beta$;
n. $\Gamma_{5F}=\max\{(\Gamma_9+2\mu+\alpha), (\Gamma_5+a_1)\}$;
o. $\Gamma_{8F}=\max\{(\Gamma_{5F}+\mu+\lambda), (\Gamma_8+a_2)\}$;
p. $\Gamma_{10}=\Gamma_{8F}+\lambda$;
q. $\Gamma_{9F}=\max\{(\Gamma_{10}+\mu), (\Gamma_9+a_3)\}$;
r. $\Gamma_{11}=\Gamma_{9F}+\lambda$;
s. $\Gamma_{10F}=\max\{(\Gamma_{11}+\mu), (\Gamma_{10}+a_2)\}$;
t. $\Gamma_{11F}=\max\{(\Gamma_{10F}+\lambda+\mu), (\Gamma_{11}+a_3)\}$;
u. $\Gamma_{12}=\max\{(\Gamma_{5F}+\lambda+a_1), (\Gamma_{11F}+\lambda+2\mu+\alpha+\beta)\}$;
v. $\Gamma_{13}=\Gamma_{12}+\lambda$; and
w. Stop;

where Γ_k denotes a time from the initial state at Node 1 to the steady state at Node k;

wherein the computed Γ_{13} is the time to reach the steady sate from the initial state.

3. The method of claim **1**, wherein the time to reach the steady sate from the initial state is computed by one of the algorithms of the transient process as follows:

a. $\Gamma_2=\alpha+\mu+\beta$;
b. $\Gamma_{2F}=\Gamma_2+\max\{a_1, (2\mu+\alpha)\}$;
c. $\Gamma_3=\Gamma_{2F}+\lambda$;
d. $\Gamma_4=\Gamma_3+\mu+\beta$;
e. $\Gamma_{3F}=\max\{(\Gamma_3+a_1), (\Gamma_4+2\mu+\alpha)\}$;
f. $\Gamma_5=\Gamma_{3F}+\lambda$;

g. $\Gamma_{4F}=\max\{(\Gamma_4+a_2), (\Gamma_{3F}+\lambda+\mu)\}$;

h. $\Gamma_6=\Gamma_{4F}+\lambda$;

i. $\Gamma_7=\Gamma_6+\mu+\beta$;

j. $\Gamma_{6F}=\max\{(\Gamma_6+a_2), (\Gamma_7+\mu)\}$;

k. $\Gamma_{7F}=\max\{(\Gamma_7+a_3), (\Gamma_{6F}+\mu+\alpha)\}$;

l. $\Gamma_8=\Gamma_{7F}+\lambda$;

m. $\Gamma_9=\Gamma_8+\mu+\beta$;

n. $\Gamma_{5F}=\max\{(\Gamma_5+a_1), (\Gamma_9+2\mu+\alpha)\}$;

o. $\Gamma_{9F}=\max\{(\Gamma_9+a_2), (\Gamma_{5F}+\mu+\lambda)\}$;

p. $\Gamma_{10}=\Gamma_{9F}+\lambda$;

q. $\Gamma_{8F}=\max\{(\Gamma_8+a_3), (\Gamma_{10}+\mu)\}$;

r. $\Gamma_{11}=\Gamma_{8F}+\lambda$;

s. $\Gamma_{10F}=\max\{(\Gamma_{10}+a_2), (\Gamma_{11}+\mu)\}$;

t. $\Gamma_{11F}=\max\{(\Gamma_{11}+a_3), (\Gamma_{10F}+\mu+\lambda)\}$;

u. $\Gamma_{12}=\max\{(\Gamma_{5F}+\lambda+a_1), (\Gamma_{11F}+\lambda+2\mu+\alpha+\beta)\}$;

v. $\Gamma_{13}=\Gamma_{12}+\lambda$; and

w. Stop;

where Γ_k denotes a time from the initial state at Node 1 to the steady state at Node k;

wherein the computed Γ_{13} is the time to reach the steady sate from the initial state.

4. A non-transitory computer-readable medium whose contents cause a computing system to perform a computer implemented method for scheduling a dual-armed cluster tools with wafer revisiting for a transient process of a wafer fabrication system, the dual-armed cluster tool comprising a robot having two arms, and three process modules PMs, each for performing a wafer-processing Step, where PM_i is used for performing a processing Step i of the three wafer-processing steps for each wafer, $i\in N_3=\{1, 2, 3\}$, the wafer revisiting comprising a wafer flow pattern as $(PM_1, (PM_2, PM_3)^2)$ with $(PM_2, PM_3)^2$ being a 2-revisiting process, the method comprising:

obtaining, by a processor, a wafer processing time a_i for each of the three processing Step $i\in N_3=\{1, 2, 3\}$, a wafer loading time β for the robot, a wafer unloading time α for the robot, a moving time μ for the robot moving from one process module to another, and a swapping time λ for a swap operation executed by the robot as follows:

holding a wafer in one arm;

unloading a processed wafer from PM_i by the other arm;

rotating; and

loading the wafer into PM_i;

determining, by a processor, a plurality of algorithms of the transient process based on a plurality of robot task sequences θs for the robot, a plurality of system states M_j, j being a positive integer, and a program evaluation and review technique, wherein the transient process starts from an initial state among the system states and reaches to a steady state among the system states, and the initial state represents that no wafer is processed in the three PMs and the arms of the robot are empty, and the final states represents that a 3rd, a 1st and a 2nd wafers are processed in PM_1, PM_2, and PM_3 for their 1st, 4th, and 3rd operations respectively, wherein each of the system states represents a Node k, k being a positive integer, in the program evaluation and review technique, wherein the algorithms of the transient process execute a plurality of if-then instructions as follows:

if $\Pi_1\leq\Pi_{local}+\psi_1$, then a cycle time of the wafer fabrication system is a constant, and

$$\Pi_{(8+3i)(11+3i)}=\begin{cases}2\Pi_{local}, \text{ if } \Pi_{local}\geq\psi_1\\ \Pi_{local}+\psi_1, \text{ if } \psi_1>\Pi_{local}\geq\varphi_1\\ \psi_1+\varphi_1, \text{ if } \varphi_1>\Pi_{local}\end{cases};$$

if $\Pi_1>2\Pi_{local}$ and $\Pi_{local}\geq\psi_1$, then a cycle time of the wafer fabrication system is Π_1 time units; and

if $\Pi_1>\Pi_{local}+\psi_1$ and $\phi_1\leq\Pi_{local}<\psi_1$, then a cycle time of the wafer fabrication system is Π_1 time units;

where $\Pi_i=a_i+\lambda$ denote a time needed for completing a wafer at the processing Step i;

$\Pi_{local}=\max\{\Pi_2, \Pi_3\}$;

$\psi_1=3\lambda+4\mu+\alpha+\beta$;

Π_{ij} denotes a time taken by θ_{ij}, where θ_{ij}, $j\geq0$, denotes a robot task sequence which transfers M_i to M_j; and

$\phi_1=2\mu+2\lambda$ denotes a local robot cycle time;

computing, by a processor, a time to reach the steady state from the initial state via performing each of the algorithms of the transient process based on the wafer processing time, the wafer loading time, the wafer unloading time, the moving time, and the swapping time; and

determining, by a processor, a corresponding robot task sequences given by the one of the algorithms of the transient process having a minimal time to reach the steady state among the computed times, wherein the determined robot task sequences are used for scheduling the transient process.

5. The non-transitory computer-readable medium of claim **4**, wherein the time to reach the steady sate from the initial state is computed by one of the algorithms of the transient process as follows:

a. $\Gamma_2=\alpha+\mu+\beta$;

b. $\Gamma_{2F}=\Gamma_2+\max\{a_1, (2\mu+\alpha)\}$;

c. $\Gamma_3=\Gamma_{2F}+\lambda$;

d. $\Gamma_4=\Gamma_3+\mu+\beta$;

e. $\Gamma_{3F}=\max\{(\Gamma_3+a_1), (\Gamma_4+2\mu+\alpha)\}$;

f. $\Gamma_5=\Gamma_{3F}+\lambda$;

g. $\Gamma_{4F}=\max\{(\Gamma_4+a_2), (\Gamma_{3F}+\lambda+\mu)\}$;

h. $\Gamma_6=\Gamma_{4F}+\lambda$;

i. $\Gamma_7=\Gamma_6+\mu+\beta$;

j. $\Gamma_{7F}=\Gamma_7+a_3$;

k. $\Gamma_{6F}=\max\{(\Gamma_{7F}+\alpha+\mu), (\Gamma_6+a_2)\}$;

l. $\Gamma_8=\Gamma_{6F}+\lambda$;

m. $\Gamma_9=\Gamma_8+\mu+\beta$;

n. $\Gamma_{5F}=\max\{(\Gamma_9+2\mu+\alpha), (\Gamma_5+a_1)\}$;

o. $\Gamma_{8F}=\max\{(\Gamma_{5F}+\mu+\lambda), (\Gamma_8+a_2)\}$;

p. $\Gamma_{10}=\Gamma_{8F}+\lambda$;

q. $\Gamma_{9F}=\max\{(\Gamma_{10}+\mu), (\Gamma_9+a_3)\}$;

r. $\Gamma_{11}=\Gamma_{9F}+\lambda$;

s. $\Gamma_{10F}=\max\{(\Gamma_{11}+\mu), (\Gamma_{10}+a_2)\}$;

t. $\Gamma_{11F}=\max\{(\Gamma_{10F}+\lambda+\mu), (\Gamma_{11}+a_3)\}$;

u. $\Gamma_{12}=\max\{(\Gamma_{5F}+\lambda+a_1), (\Gamma_{11F}+\lambda+2\mu+\alpha+\beta)\}$;

v. $\Gamma_{13}=\Gamma_{12}+\lambda$; and

w. Stop;

where Γ_k denotes a time from the initial state at Node 1 to the steady state at Node k;

wherein the computed Γ_{13} is the time to reach the steady sate from the initial state.

6. The non-transitory computer-readable medium of claim **4**, wherein the time to reach the steady sate from the initial state is computed by one of the algorithms of the transient process as follows:

a. $\Gamma_2=\alpha+\mu+\beta$;

b. $\Gamma_{2F}=\Gamma_2+\max\{a_1, (2\mu+\alpha)\}$;

c. $\Gamma_3=\Gamma_{2F}+\lambda$;

d. $\Gamma_4=\Gamma_3+\mu+\beta$;

e. $\Gamma_{3F}=\max\{(\Gamma_3+a_1), (\Gamma_4+2\mu+\alpha)\}$;

f. $\Gamma_5=\Gamma_{3F}+\lambda$;

g. $\Gamma_{4F}=\max\{(\Gamma_4+a_2), (\Gamma_{3F}+\lambda+\mu)\}$;

h. $\Gamma_6=\Gamma_{4F}+\lambda$;

i. $\Gamma_7=\Gamma_6+\mu+\beta$;

j. $\Gamma_{6F}=\max\ \{(\Gamma_6+a_2), (\Gamma_7+\mu)\}$;

k. $\Gamma_{7F}=\max\{(\Gamma_7+a_3), (\Gamma_{6F}+\mu+\alpha)\}$;

l. $\Gamma_8=\Gamma_{7F}+\lambda$;

m. $\Gamma_9=\Gamma_8+\mu+\beta$;

n. $\Gamma_{5F}=\max\{(\Gamma_5+a_1), (\Gamma_9+2\mu+\alpha)\}$;

o. $\Gamma_{9F}=\max\{(\Gamma_9+a_2), (\Gamma_{5F}+\mu+\lambda)\}$;

p. $\Gamma_{10}=\Gamma_{9F}+\lambda$;

q. $\Gamma_{8F}=\max\{(\Gamma_8+a_3), (\Gamma_{10}+\mu)\}$;

r. $\Gamma_{11}=\Gamma_{8F}+\lambda$;

s. $\Gamma_{10F}=\max\{(\Gamma_{10}+a_2), (\Gamma_{11}+\mu)\}$;

t. $\Gamma_{11F}=\max\{(\Gamma_{11}+a_3), (\Gamma_{10F}+\mu+\lambda)\}$;

u. $\Gamma_{12}=\max\{(\Gamma_{5F}+\lambda+a_1), (\Gamma_{11F}+\lambda+2\mu+\alpha+\beta)\}$;

v. $\Gamma_{13}=\Gamma_{12}+\lambda$; and

w. Stop;

where Γ_k denotes a time from the initial state at Node 1 to the steady state at Node k;

wherein the computed Γ_{13} is the time to reach the steady sate from the initial state.

7. The method of claim **1**, wherein the algorithms of the transient process further execute a plurality of if-then instructions as follows:

if $\Pi_1\leq\Pi_{local}+\psi_1$, then $\omega_1^{8+3i}=0$ holds, $i\geq0$;

if $2\Pi_{local}\geq\Pi_1>\Pi_{local}+\psi_1$ and $\Pi_{local}\geq\psi_1$, then a cycle time of the wafer fabrication system is $2\Pi_{local}$ time units; and

if $\Pi_1>\Pi_{local}+\psi_1$ and $\Pi_{local}<\phi_1$, then a cycle time of the wafer fabrication system is

$$\Pi_{(8+3i)(11+3i)} = \begin{cases} \Pi_1, \text{ if } \Pi_1 > \varphi_1 + \psi_1 \\ \varphi_1 + \psi_1, \text{ if } \Pi_1 \leq \varphi_1 + \psi_1 \end{cases};$$

where ω_i^j denotes a robot waiting time before the robot unloads a wafer from the PM_i during a period from states M_{j-1} to M_j, $j>0$.

8. The non-transitory computer-readable medium of claim **4**, wherein the algorithms of the transient process further execute a plurality of if-then instructions as follows:

if $\Pi_1\leq\Pi_{local}+\psi_1$, then $\omega_1^{8+3i}=0$ holds, $i\geq0$;

if $2\Pi_{local}\geq\Pi_1>\Pi_{local}+\psi_1$ and $\Pi_{local}\geq\psi_1$, then a cycle time of the wafer fabrication system is $2\Pi_{local}$ time units; and

if $\Pi_1>\Pi_{local}+\psi_1$ and $\Pi_{local}<\phi_1$, then a cycle time of the wafer fabrication system is

$$\Pi_{(8+3i)(11+3i)} = \begin{cases} \Pi_1, \text{ if } \Pi_1 > \varphi_1 + \psi_1 \\ \varphi_1 + \psi_1, \text{ if } \Pi_1 \leq \varphi_1 + \psi_1 \end{cases};$$

where ω_i^j denotes a robot waiting time before the robot unloads a wafer from the PM_i during a period from states M_{j-1} to M_j, $j>0$.

* * * * *